Blue Coat® Systems ProxySG® Appliance
Command Line Interface Reference

Version SGOS 6.5.2.10

BLUE COAT

Contact Information

Americas:
Blue Coat Systems Inc.
420 North Mary Ave
Sunnyvale, CA 94085-4121

Rest of the World:
Blue Coat Systems International SARL
3a Route des Arsenaux
1700 Fribourg, Switzerland

http://www.bluecoat.com/contact/customer-support

http://www.bluecoat.com

For concerns or feedback about the documentation:
documentation@bluecoat.com

Americas:

Blue Coat Systems, Inc.
420 N. Mary Ave.
Sunnyvale, CA 94085

Rest of the World:

Blue Coat Systems International SARL
3a Route des Arsenaux
1700 Fribourg, Switzerland

Document Number: 231-03035

Document Revision: SGOS 6.5.2.10—09/2014

Contents

Chapter 3: Privileged Mode Configure Commands

Chapter 1: Introduction

To help you configure and manage your Blue Coat ProxySG appliance, Blue Coat developed a software suite that includes an easy-to-use graphical interface called the Management Console and a Command Line Interface (CLI). The CLI allows you to perform the superset of configuration and management tasks; the Management Console, a subset.

This reference guide describes each of the commands available in the CLI.

Audience for this Document

This reference guide is written for system administrators and experienced users who are familiar with network configuration. Blue Coat assumes that you have a functional network topography, that you and your Blue Coat Sales representative have determined the correct number and placement of the ProxySG, and that those appliances have been installed in an equipment rack and at least minimally configured as outlined in the Blue Coat *Installation Guide* that accompanied the device.

Organization of this Document

This document contains the following chapters:

Chapter 1 – Introduction

The organization of this document; conventions used; descriptions of the CLI modes; and instructions for saving your configuration.

Chapter 2 – Standard and Privileged Mode Commands

All of the standard mode commands, including syntax and examples, in alphabetical order. All of the privileged mode commands (except for the `configure` commands, which are described in Chapter 3), including syntax and examples, in alphabetical order.

Chapter 3 – # Configure Mode Commands

The `#configure` command is the most used and most elaborate of all of the CLI commands.

Other Blue Coat Documentation

Access current SGOS documentation at Blue Touch Online (BTO):

https://bto.bluecoat.com/documentation/pubs/ProxySG

The following documentation is available at BTO:

- Blue Coat SGOS Release Notes
- Blue Coat SGOS Upgrade/Downgrade Guide
- Blue Coat SGOS Administration Guide
- Blue Coat SGOS Visual Policy Manager Reference (includes some advanced policy tasks)
- Blue Coat SGOS Content Policy Language Reference

Blue Coat also provides various other deployment guides targeted for specific solutions.

Document Conventions

The following table lists the typographical and CLI syntax conventions used in this manual.

Table 1-1. Document Conventions

Conventions	Definition
Italics	The first use of a new or Blue Coat-proprietary term.
`Courier font`	Screen output. For example, command line text, file names, and Blue Coat Content Policy Language (CPL).
`Courier Italics`	A command line variable that is to be substituted with a literal name or value pertaining to the appropriate facet of your network system.
`Courier Boldface`	A Blue Coat literal to be entered as shown.
Arial Boldface	Screen elements in the Management Console.
`{ }`	One of the parameters enclosed within the braces must be supplied
`[ ]`	An optional parameter or parameters.
`|`	Either the parameter before or after the pipe character can or must be selected, but not both.

Notes and Warnings

The following is provided for your information and to caution you against actions that can result in data loss or personal injury:

Note: Information to which you should pay attention.

Important: Critical information that is not related to equipment damage or personal injury (for example, data loss).

WARNING: Used *only* to inform you of danger of personal injury or physical damage to equipment. An example is a warning against electrostatic discharge (ESD) when installing equipment.

SSH and Script Considerations

Consider the following when using the CLI during an SSH session or in a script:

Case Sensitivity. CLI command literals and parameters are not case-sensitive.

Command Abbreviations. You can abbreviate CLI commands, provided you supply enough command characters as to be unambiguous. For example:

```
SGOS# configure terminal
```

The preceding can be shortened to:

```
SGOS# conf t
```

Standard and Privileged Modes

The ProxySG CLI has three major modes—*standard*, *privileged*, and *configure privileged*. In addition, privileged mode has several subordinate modes. See the introduction in Chapter 2: "Standard and Privileged Mode Commands" on page 13 for details about the different modes.

- Standard mode prompt: `>`
- Privileged mode prompt: `#`
- Configure Privileged mode prompt: `#(config)`

Accessing Quick Command Line Help

You can access command line help at any time during a session. The following commands are available in both standard mode and privileged mode.

To access a comprehensive list of mode-specific commands:

Enter `help` or `?` at the prompt.

The `help` command displays how to use CLI help. For example:

```
SGOS> help
Help may be requested at any point in a command
by typing a question mark '?'.
1. For a list of available commands, enter '?' at
   the prompt.
2. For a list of arguments applicable to a command,
   precede the '?' with a space (e.g. 'show ?')
3. For help completing a command, do not precede
   the '?' with a space (e.g. 'sh?')
```

The `?` command displays the available commands. For example:

```
SGOS> ?
  display                 Display a text based url
  enable                  Turn on privileged commands
  exit                    Exit command line interface
  help                    Information on help
  ping                    Send echo messages
  show                    Show running system information
  traceroute              Trace route to destination
```

To access a command-specific parameter list:

Enter the command name, followed by a space, followed by a question mark.

You must be in the correct mode—standard or privileged—to access the appropriate help information. For example, to get command completion help for `pcap`:

```
SGOS# pcap ?
bridge                       Setup the packet capture mode for bridges
filter                      Setup the current capture filter
```

To get command completion for configuring the time:

```
SGOS#(config) clock ?
day                          Set UTC day
hour                         Set UTC hour
```

To access the correct spelling and syntax, given a partial command:

Type the first letter, or more, of the command, followed by a question mark (no spaces).

You must be in the correct mode—standard or privileged—to access the appropriate help information. For example:

```
SGOS# p?
pcap   ping   purge-dns-cache
```

Chapter 2: Standard and Privileged Mode Commands

This chapter describes and provides examples for the Blue Coat ProxySG appliance standard and privileged mode CLI commands. These modes have fewer permissions than enabled mode commands.

This chapter includes information about the following topics:

Standard Mode Commands

Standard mode is the default mode when you first log in to the CLI. From standard mode, you can view but not change configuration settings. This mode can be password protected, but it is not required.

The standard mode prompt is a greater-than sign; for example:

```
ssh> ssh -l username IP_address
password: ******
SGOS>
```

> display

Synopsis

Use this command to display the content (such as HTML or Javascript) for the specified URL. This content is displayed one screen at a time. "—More—" at the bottom of the terminal screen indicates that there is additional content. Press the Spacebar to display the next batch of content; press the Enter key to display one additional line of content.

This command is used for general HTTP connectivity testing

Syntax

> **display** *url*

where *url* is a valid, fully-qualified text Web address.

Example

```
SGOS> display http://www.bluecoat.com
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<HTML>
<HEAD>
<TITLE>Blue Coat Systems</TITLE>
<META http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
<META NAME="keywords" CONTENT="spyware WAN application spyware removal spy ware
spyware remover application delivery to branch office accelerate performance
applications remove spyware spyware application delivery secure application
acceleration control SSL threat anti-virus protection WAN optimization AV
appliance spyware blocker application acceleration distributed security
application performance spyware killer spyware WebFilter protection CIFS MAPI
streaming video Web application security branch offices secure endpoint
protection SSL policy control remote user acceleration WAN delivery application
performance WebFilter endpoint security fast WAN policy control spyware detection
spyware eliminator block endpoint security spyware secure MAPI appliances SSL AV
policy control stop spyware remove AV appliance SSL proxy Http secure Web
application acceleration encryption Proxy Internet Proxy Internet Proxy Cache
security proxy cache proxy server CIFS proxy servers branch office Web proxy
appliance enterprise data center accelerate WAN and CIFS and MAPI and streaming
video policy protection blue coat Web proxy Internet Web AV security systems blue
coat branch office anti-virus performance blue coat remote users WAN performance
acceleration Internet MAPI monitoring AV endpoint Internet application delivery
management endpoint protection and security and acceleration of application
content delivery with policy control Internet CIFS Web application filtering
content filtering Web filtering web filter WAN filtered internet application
acceleration">

    .
    .
    .
```

> enable

Synopsis

Use this command to enter Privileged mode. Privileged mode commands enable you to view and change your configuration settings. A password is always required.

Syntax

```
> enable
```

The `enable` command has no parameters or subcommands.

For More Information

- **# disable** on page 49
- **#(config) security password and hashed_password** on page 346
- **#(config) security username** on page 369

Example

```
SGOS> enable
Enable Password:******
SGOS# conf t
SGOS(config)
```

where `conf t` is a shortcut to typing `configure terminal`.

> exit

Synopsis

Use this command to exit the CLI. In privileged and configuration mode, `exit` returns you to the previous prompt.

Syntax

```
> exit
```

The `exit` command has no parameters or subcommands.

Example

```
SGOS> exit
```

> help

See Accessing Quick Command Line Help on page 11 for information about this command.

> ping

Synopsis

Use this command to verify whether an Internet Protocol version 4 (IPv4) host is reachable across a network.

Syntax

```
> ping {IPv4 address | hostname}
```

Subcommands

> ping *IPv4 address*
Specifies the IPv4 address you want to verify.

> ping *hostname*
Specifies the name of the host you want to verify.

Example

```
SGOS> ping 10.25.36.47
Type escape sequence to abort.
Sending 5, 64-byte ICMP Echos to 10.25.36.47, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5),
round-trip min/avg/max = 0/0/1 ms
Number of duplicate packets received = 0
```

> ping6

Synopsis

Use this command to verify whether an Internet Protocol version 6 (IPv6) host is reachable across a network.

Syntax

```
> ping6 {IPv6 address | hostname}
```

Subcommands

> ping6 *IPv6 address*
Specifies the IPv6 address you want to verify.

> ping6 *hostname*
Specifies the name of the host you want to verify.

Example

```
SGOS> ping6 fe80::2d0:83ff:fe05:780%0:0
PING6(56=40+8+8 bytes) fe80::2d0:83ff:fe05:780 --> fe80::2d0:83ff:fe05:780%0:0
16 bytes from fe80::2d0:83ff:fe05:780%0:0, icmp_seq=0 hlim=64 time=0.799 ms
16 bytes from fe80::2d0:83ff:fe05:780%0:0, icmp_seq=1 hlim=64 time=0.761 ms
16 bytes from fe80::2d0:83ff:fe05:780%0:0, icmp_seq=2 hlim=64 time=1.630 ms
16 bytes from fe80::2d0:83ff:fe05:780%0:0, icmp_seq=3 hlim=64 time=1.703 ms
16 bytes from fe80::2d0:83ff:fe05:780%0:0, icmp_seq=4 hlim=64 time=3.745 ms

--- fe80::2d0:83ff:fe05:780%0:0 ping6 statistics ---
5 packets transmitted, 5 packets received, 0.0% packet loss
round-trip min/avg/max/std-dev = 0.761/1.728/3.745/1.084 ms
```

> ping6

Synopsis

Use this command to verify whether a particular host is reachable across a network.

Syntax

```
> ping6 {ipv6_address | hostname}
```

Subcommands

> ping6 *hostname*
Specifies the name of the host you want to verify.

> ping6 *ipv6_address*
Specifies the IPv6 address you want to verify.

Example

```
SGOS> ping6 805B:2D9D:DC28:0000:0000:FC57:D4C8:1FFF
% ping6[65]: UDP connect
PING6(56=40+8+8 bytes) :: --> 2001:DB8::/32
Success rate is 100 percent (5/5),
round-trip min/avg/max = 0/0/1 ms
Number of duplicate packets received = 0
```

> show

Synopsis

Use this command to display system information. You cannot view all show commands, here, only those available in the standard mode. You must be in privileged mode to show all available commands.

Syntax

> show [*subcommands*]

Subcommands

Note: Click subcommand links for additional information.

> show accelerated-pac
Displays accelerated PAC file information.

> show access-log on page 26
Displays the current access log settings.

> show advanced-url
Displays the advanced URL for statistics...

> **show appliance-name**
Displays the name of the appliance.

> show arp-table
Displays TCP/IP ARP table information.

> show bandwidth-gain
Displays bandwidth gain status, mode, and the status of the "substitute get for get-if-modified-since," "substitute get for HTTP 1.1 conditional get," and "never refresh before specified object expiry" features.

> show bandwidth-management on page 27
Displays bandwidth management configuration and statistics information.

> show bridge on page 28
Displays information about bridging on the system.

> show cachepulse
Displays CachePulse service settings.

> show caching
Displays data regarding cache refresh rates and settings and caching policies.

> show cifs on page 29
Displays Common Internet File System (CIFS) information

> show clock
Displays the current ProxySG time setting.

> show commands on page 30
Displays the available CLI commands.

> show content-distribution
Displays the average sizes of objects in the cache.

> show cpu
Displays CPU usage.

> show cpu-monitor
Displays the state of the CPU monitor.

> show diagnostics on page 31
Displays remote diagnostics information.

> show disk on page 32
Displays disk information, including slot number, vendor, product ID, revision and serial number, capacity, and status, about all disks or a specified disk.

> show dns
Displays primary and alternate DNS server data.

> **show dns-forwarding**
Displays the DNS servers and the imputing name.

> show download-paths
Displays downloaded configuration path information, including the policy list, accelerated PAC file, HTTP error page, RIP settings, static route table, upgrade image, and WCCP settings.

> show epmapper [statistics]
Displays proxy settings or statistics.

> show event-log [configuration]
Show the event-log configuration.

> show exceptions on page 33
Displays all exceptions or just the built-in or user-defined exception you specify.

> show external-services [statistics]
Displays external services or external services statistics information.

> show failover [***group_address***]
Displays failover settings for the specified group or all groups.

> show forwarding
Displays advanced forwarding settings, including download-via-forwarding, health check, and load balancing status, and the definition of forwarding hosts/groups and advanced forwarding rules.

> show ftp
Displays the FTP settings on the system.

> **show general**
Displays the general settings.

> **show geolocation**
Displays geolocation settings.

> show health-checks
Displays health check information.

> show http
Displays HTTP configuration information.

> show http-stats
Displays HTTP statistics, including HTTP statistics version number, number of connections accepted by HTTP, number of persistent connections that were reused, and the number of active client connections.

> show identd
Displays IDENTD service settings.

> show im on page 34
Displays IM information

> show installed-systems
Displays ProxySG system information, listing the current five version and release numbers, boot and lock status, and timestamp information.

> show interface {all | *interface_number*}
Displays interface status and configuration information.

> show ip-default-gateway
Specifies the default IP gateway.

> show ip-route-table
Displays route table information.

> show ip-stats on page 35
Displays TCP/IP statistics

> show ipv6
Displays current settings for IPv6-related options (bypass IPv6 traffic, auto-linklocal, forwarding).

> show licenses
Displays license information.

> show management-services
Displays information about the management services enabled or disabled on the system.

> show mapi
Displays settings for the MAPI proxy.

> show ndp
Shows TCP/IP Neighbor Discovery Protocol (NDP) table. NDP performs functions for IPv6 similar to ARP for IPv4.

> show netbios
Displays NETBIOS settings.

> show netflow
Displays NetFlow settings. The `#(config netflow) view` command also displays these settings. For more information, see **#(config) netflow** on page 249.

> show ntp
Displays NTP servers status and information.

> show p2p [statistics]
Displays P2P statistics.

> show policy [listing | order |policy]
Displays current state of the policy.

> show private-network
Displays the private network subnets and domains.

> show profile
Displays the system profile.

> show proxy-client
Displays the proxy client settings.

> show proxy-services
Displays information about proxy services.

> show reflect-client-ip
Displays the client IP reflection.

> show resources
Displays allocation of disk and memory resources.

> show restart
Displays system restart settings, including core image information and compression status.

> show return-to-sender
Displays "return to sender" inbound and outbound settings.

> show rip {default-route | parameters| routes | statistics}
Displays information on RIP settings, including parameters and configuration, RIP routes, and RIP statistics.

> show services
Displays information about services.

> show service-groups
Displays proxy service groups.

> show sessions
Displays information about the CLI session.

> show shell
Displays the settings for the shell, including the maximum connections, the prompt, and the realm- and welcome-banners.

> show smtp
Displays SMTP configuration, including the server domain name or IP address, port number, and sender's email address.

> show snmp
Displays SNMP statistics, including status and MIB variable and trap information

> show socks-gateways
Displays SOCKS gateway settings.

> show socks-machine-id
Displays the identification of the secure sockets machine.

> show socks-proxy
Displays SOCKS proxy settings.

> show sources on page 36
Displays source listings for installable lists, such as the license key, policy files, RIP settings, static route table, and WCCP settings files.

> show ssl on page 37
Displays ssl settings.

> show static-routes
Displays static route table information.

> show status
Displays current system status information, including configuration information and general status information.

> show streaming on page 38
Displays QuickTime, RealNetworks, or Microsoft Windows Media information, and client and total bandwidth configurations and usage.

> show tcp-ip
Displays TCP-IP parameters.

> show terminal
Displays terminal configuration parameters and subcommands.

> show timezones
Displays timezones used.

> show trust-destination-ip
Displays the trust destination IP

> show user-overflow-action
Displays the user overflow action.

> show version
Displays ProxySG hardware and software version and release information and backplane PIC status.

> show virtual-ip
Displays the current virtual IP addresses

> show wccp {**configuration** | **statistics** | **status**}
Displays WCCP configuration and statistics information. You can also view WCCP service-group information.

> show xml-config
Displays the registry settings.

Examples

```
SGOS> show caching
Refresh:
     Estimated access freshness is 100.0%
     Let the ProxySG Appliance manage refresh bandwidth
     Current bandwidth used is 0 kilobits/sec
Policies:
     Do not cache objects larger than 1024 megabytes
     Cache negative responses for 0 minutes
     Let the ProxySG Appliance manage freshness
FTP caching:
     Caching FTP objects is enabled
     FTP objects with last modified date, cached for 10% of last modified time
     FTP objects without last modified date, initially cached for 24 hours
SGOS> show resources
Disk resources:
     Maximum objects supported:   1119930
     Cached Objects:              0
     Disk used by system objects: 537533440
     Disk used by access log:     0
     Total disk installed:        18210036736
Memory resources:
     In use by cache:             699203584
     In use by system:            83230176
     In use by network:           22872608
     Total RAM installed:         805306368
SGOS> show failover configuration group_address
Failover Config
Group Address: 10.25.36.47
     Multicast Address    : 224.1.2.3
     Local Address        : 10.9.17.159
     Secret               : none
     Advertisement Interval: 40
     Priority             : 100
     Current State        : DISABLED
     Flags                : V M
```

Three flags exist, set as you configure the group.

V—Specifies the group name is a virtual IP address.

R—Specifies the group name is a physical IP address

M—Specifies this machine can be configured to be the master if it is available

> show access-log

Synopsis

Displays the current access log settings.

Syntax

> **show access-log** [*subcommands*]

Subcommands

> **show access-log default-logging**
Display the access log default policy.

> show access-log format brief
Displays the access log format names.

> show access-log format *format_name*
Displays the access log with the specified *format_name*.

> show access-log format
Displays the access-log formats for all log types.

> show access-log log brief
Displays the access log names.

> show access-log log *log_name*
Displays the access log with the specified *log_name*.

> show access-log log
Displays the access-log for all logs.

> show access-log statistics *log_name*
Displays access-log statistics for the specific *log_name.*

> show access-log statistics
Displays all access-log statistics.

For More Information

- "Creating Custom Access Log Formats" in *SGOS 6.5.x Administration Guide*

Example

```
> show access-log format brief
Formats:
squid
ncsa
main
im
streaming
surfcontrol
surfcontrolv5
p2p
ssl
cifs
mapi
```

> show bandwidth-management

Synopsis

Displays the bandwidth management state (enabled or disabled) or statistics.

Syntax

```
> show bandwidth-management {configuration | statistics}
```

Subcommands

> show bandwidth-management configuration *bandwidth_class*
Displays the bandwidth-management configuration for the specified bandwidth class . If you do not specify a bandwidth class, displays the bandwidth-management configuration for the system.

> show bandwidth-management statistics *bandwidth_class*
Displays the bandwidth-management statistics for the specified bandwidth class. If you do not specify a bandwidth class, displays the bandwidth-management statistics for the system.

For More Information

- *SGOS 6.5.x Administration Guide*

Example

```
> show bandwidth-management configuration
Bandwidth Management Enabled
```

> show bridge

Synopsis

Displays bridge configuration and statistics.

Syntax

```
> show bridge [subcommands]
```

Subcommands

> **show bridge configuration** ***[bridge_name]***
Displays the bridge configuration for the specified ***bridge_name*** or for all interfaces on the system.

> **show bridge fwtable** ***[bridge_name]***
Displays the bridge forwarding table for the specified ***bridge_name*** or for all interfaces on the system.

> **show bridge statistics** ***[bridge_name]***
Displays the bridge statistics for the specified ***bridge_name*** or for all interfaces on the system.

For More Information

- *SGOS 6.5.x Administration Guide*

Example

```
> show bridge configuration
Bridge passthru-0 configuration:
  Interface 0:0
    Internet address: 10.9.59.246
    Internet subnet:  255.255.255.0
    MTU size:         1500
    Spanning tree:    disabled
    Allow intercept:  enabled
    Reject inbound:   disabled
    Status:           autosensed full duplex, 100 megabits/sec network
  Interface 0:1
    MTU size:         1500
    Spanning tree:    disabled
    Allow intercept:  enabled
    Reject inbound:   disabled
    Status:           autosensed no link
```

> show cifs

Synopsis

Show Common Internet File System (CIFS) information.

Syntax

```
> show cifs [subcommands]
```

Subcommands

> show cifs configuration

Displays the CIFS configuration settings, such as whether read-ahead is enabled/disabled and write-back is full or none. For more information on these settings, see **#(config) cifs** on page 147.

> show cifs directory *url*

Displays contents of the specified directory located in the ProxySG cache. *url* is in the format:
`cifs://server/share/path-to-file`

To enter file or directory names that contain spaces, substitute each space with the following escape code: `%20`. If the *path-to-file* contains a percent sign, substitute the `%` with `%25`.
The output lists each file or directory name, the date and time the file/directory was last updated, and the file size.

Note that you must be in enable mode to use the **show cifs directory** command.

> show cifs file *url*

Displays information about a specific CIFS file or directory located in the ProxySG cache. *url* is in the format:
`cifs://server/share/path-to-file`

To enter file or directory names that contain spaces, substitute each space with the following escape code: `%20`. If the *path-to-file* contains a percent sign, substitute the `%` with `%25`.

The output lists the object size, and when the file/directory was created, last accessed, and last modified.

Note that you must be in enable mode to use the **show cifs file** command.

> show cifs statistics

Displays statistics about CIFS read/write file operations.

Example

```
> show cifs file cifs://atlas/public/prepop/a/a1.txt
;
cifs://atlas/public/prepop/a/a1.txt

Type: file
Cached object size: 409,121
Data bytes in cache: 409,121

Creation Time: Thu, 09 Jul 2009 03:36:15 UTC
Last Access Time: Wed, 14 Oct 2009 17:36:25 UTC
Last Write Time: Thu, 09 Jul 2009 03:36:24 UTC
Change Time: Thu, 09 Jul 2009 03:36:24 UTC
```

> show commands

Synopsis

Displays the available CLI commands.

Syntax

```
> show commands [subcommands]
```

Subcommands

> show commands delimited [all | privileged]
Delimited displays commands so they can be parsed.

> show commands formatted [all | privileged]
Formatted displays commands so they can be viewed easily.

Example

```
> show commands formatted
1:show                          Show running system information
  2:access-log                  Access log settings
    3:log                       Show Access log configuration
      4:brief                   Show Access log names
        <log-name>
    3:format                    Show Access log format configuration
      4:brief                   Show Access log format names
        <format-name>
    3:statistics                Show Access log statistics
      <logName>
    3:default-logging           Show Access log default policy
> show commands delimited
1;show;Show running system information;sh;0;11
2;access-log;Access log settings;acces;0;11
3;log;Show Access log configuration;l;0;11
4;brief;Show Access log names;b;0;11
p;<log-name>;*;*;0;14
3;format;Show Access log format configuration;f;0;11
4;brief;Show Access log format names;b;0;11
p;<format-name>;*;*;0;14
3;statistics;Show Access log statistics;s;0;11
p;<logName>;*;*;0;14
3;default-logging;Show Access log default policy;d;0;11
```

> show diagnostics

Synopsis

Displays remote diagnostics information, including version number, and whether the Heartbeats feature and the ProxySG monitor are currently enabled.

Syntax

```
> show diagnostics [subcommands]
```

Subcommands

> show diagnostics configuration
Displays diagnostics settings.

> show diagnostics cpu-monitor
Displays the CPU Monitor results.

> show diagnostics service-info
Displays service-info settings.

> show diagnostics snapshot
Displays the snapshot configuration.

Example

```
> show diagnostics snapshot
Snapshot sysinfo
  Target:        /sysinfo
  Status:        Enabled
  Interval:      1440 minutes
  To keep:       30
  To take:       Infinite
  Next snapshot: 2006-03-18 00:00:00 UTC
Snapshot sysinfo_stats
  Target:        /sysinfo-stats
  Status:        Enabled
  Interval:      60 minutes
  To keep:       100
  To take:       Infinite
  Next snapshot: 2006-03-17 20:00:00 UTC
```

> show disk

Synopsis

Displays information about the specified hard disk(s), including slot number, drive manufacturer, product ID/model, revision and serial number, capacity, SGOS compatibility information, and disk status.

The disk status line item displays information based on the current status of the selected hard disk drive:

- `Present`

  Indicates that the hard disk drive is properly mounted and available for access by the appliance.

- `Empty`

  Indicates that the hard disk drive slot is not occupied by a disk drive.

- `Initializing`

  Indicates that the hard disk drive is in the process of being mounted for use by the appliance.

- `Offline`

  Indicates that the hard disk drive is offline and no longer usable by the appliance.

- `Error`

  Indicates that a hard drive disk is faulty.

- `Present (partition offline)`

  Indicates that a drive partition is down; however the disk drive is still active.

Syntax

```
> show disk {disk_number | all}
```

Subcommands

> show disk *disk_number*
Displays information on the specified disk.

> show disk all

Displays information about all installed disks in the ProxySG appliance.

Example

```
> show disk 1
  Disk in slot 1
  Vendor: SEAGATE
  Product: ST340014A
  Revision: 8.54
  Disk serial number: 5JVQ76VS
  Capacity: 40020664320 bytes
  Pre 6.2 compatible: yes
  Status: present (partition offline)
```

> show exceptions

Synopsis

Displays all exceptions or just built-in or user defined exceptions.

Syntax

```
> show exceptions [built-in_id | user-defined_id]
```

For More Information

- #(config) exceptions on page 188

Example

```
> show exceptions
Built-in:
authentication_failed
authentication_failed_password_expired
authentication_mode_not_supported
authentication_redirect_from_virtual_host
authentication_redirect_off_box
authentication_redirect_to_virtual_host
authentication_success
authorization_failed
bad_credentials
client_failure_limit_exceeded
configuration_error
connect_method_denied
content_filter_denied
content_filter_unavailable
dns_server_failure
dns_unresolved_hostname
dynamic_bypass_reload
gateway_error
icap_communication_error
icap_error
internal_error
invalid_auth_form
invalid_request
invalid_response
license_exceeded
license_expired
method_denied
not_implemented
notify
notify_missing_cookie
policy_denied
policy_redirect
radius_splash_page
redirected_stored_requests_not_supported
refresh
server_request_limit_exceeded
silent_denied
spoof_authentication_error
ssl_client_cert_revoked
ssl_domain_invalid
ssl_failed
ssl_server_cert_expired
ssl_server_cert_revoked
ssl_server_cert_untrusted_issuer
tcp_error
transformation_error
unsupported_encoding
unsupported_protocol
```

> show im

Synopsis

Displays Instant Messaging settings.

Syntax

```
> show im [subcommands]
```

Subcommands

> show im configuration
Displays IM configuration information.

> show im aol-statistics
Displays statistics of AOL IM usage.

> show im msn-statistics
Displays statistics of MSN IM usage.

> show im yahoo-statistics
Displays statistics of Yahoo! IM usage.

For More Information

- *"Managing Instant Messaging Protocols" in SGOS 6.5.x Administration Guide*

Example

```
> show im configuration
IM Configuration
aol-admin-buddy:          Blue Coat SG
msn-admin-buddy:          Blue Coat SG
yahoo-admin-buddy:        Blue Coat SG
exceptions:               out-of-band
buddy-spoof-message:      <none>
http-handoff:             enabled
explicit-proxy-vip:       <none>
aol-native-host:          login.oscar.aol.com
aol-http-host:            aimhttp.oscar.aol.com
aol-direct-proxy-host:    ars.oscar.aol.com
msn-native-host:          messenger.hotmail.com
msn-http-host:            gateway.messenger.hotmail.com
yahoo-native-host:        scs.msg.yahoo.com
yahoo-http-host:          shttp.msg.yahoo.com
yahoo-http-chat-host:     http.chat.yahoo.com
yahoo-upload-host:        filetransfer.msg.yahoo.com
yahoo-download-host:      .yahoofs.com
```

> show ip-stats

Synopsis

Displays TCP/IP statistics.

Syntax

> **show ip-stats** [*subcommands*]

Subcommands

> show ip-stats all
Display TCP/IP statistics.

> show ip-stats interface {all | *number***}**
Displays TCP/IP statistics for all interfaces or for the specified number (0 to 7).

> show ip-stats ip
Displays IP statistics.

> show ip-stats memory
Displays TCP/IP memory statistics.

> show ip-stats summary
Displays TCP/IP summary statistics.

> show ip-stats tcp
Displays TCP statistics.

> show ip-stats udp
Displays UDP statistics.

Example

```
> show ip-stats summary
; TCP/IP Statistics
TCP/IP General Statistics
Entries in TCP queue: 12
Maximum entries in TCP queue: 19
Entries in TCP time wait queue: 0
Maximum entries in time wait queue: 173
Number of time wait allocation failures: 0
Entries in UDP queue: 2
```

> show sources

Synopsis

Displays source listings for installable lists, such as the license key, policy files, RIP settings, static route table, and WCCP settings files.

Syntax

```
> show sources [subcommands]
```

Subcommands

> authentication-form
Displays the specified authentication form.

> show sources crl
Displays the specified CRL.

> show sources exceptions
Displays the exception code.

> show sources forwarding
Displays forwarding settings.

> show sources license-key
Displays license information

> show sources policy {central | local | forward | vpm-cpl | vpm-xml}
Displays the policy file specified.

> show sources rip-settings
Displays RIP settings.

> show sources socks-gateways
Displays the SOCKS gateways settings.

> show sources static-route-table
Displays the static routing table information.

> show sources wccp-settings
Displays WCCP settings.

Example

```
> show sources socks-gateways
# Current SOCKS Gateways Configuration
# No update
# Connection attempts to SOCKS gateways fail: closed
socks_fail closed
# 0 gateways defined, 64 maximum
# SOCKS gateway configuration
# gateway <gateway-alias> <gateway-domain> <SOCKS port>
#    [version=(4|5 [user=<user-name> password=<password>]
#    [request-compression=yes|no])]
# Default fail-over sequence.
# sequence <gateway-alias> <gateway-alias> ...
# The default sequence is empty.
# SOCKS Gateways Configuration Ends
```

> show ssl

Synopsis

Displays SSL settings

Syntax

```
> show ssl {ccl [list_name] | ssl-client [ssl_client]}
```

Subcommands

> show appliance-certificate-request
Displays the CA certificate configuration.

> show ssl ccl [*list_name*]
Displays currently configured CA certificate lists or configuration for the specified list_name.

> show ssl certificate
Displays the specified certificate configuration.

> show ssl crl
Displays information for the specified crl.

> show ssl external-certificate
Displays the specified external certificate configuration.

> show ssl keypair
Displays the specified key pair configuration.

> show ssl keyring
Displays the specified keyring configuration.

> show ssl keyring [verbose] *list_name*
Displays the keylist extractor as well as the keyring IDs and their respective extractor values. Use **verbose** to display the certificate field values of the keylist.

> show ssl ocsp
Displays the specified SSL OCSP configuration.

> show ssl proxy
Displays the SSL proxy configuration.

> show ssl signing-request
Displays the specified certificate signing request configuration.

> show ssl ssl-client [*ssl_client*]
Displays information about the specified SSL client.

> show ssl ssl-device-profile
Displays information about the specified SSL device profile.

> show ssl ssl-nego-timeout
Displays the SSL negotiation timeout configuration.

> show ssl summary
Displays the SSL summary information.

Example

```
> show ssl ssl-client
SSL-Client Name   Keyring  CCL      Protocol
---------------   ------- -----     ------------
default           <None>    browser-trusted   tlsv1 tlsv1.1 tlsv1.2
```

> show streaming

Synopsis

Displays QuickTime, Real Media, Windows Media, Flash, Apple HLS, Adobe HDS, or Microsoft Smooth configurations and statistics.

Syntax

```
> show streaming [subcommands]
```

Subcommands

> show streaming configuration
Displays global streaming configuration.

> show streaming adobe-hds {configuration}
View the current Adobe HDS configuration.

> show streaming apple-hls {configuration}
View the current Apple HLS configuration.

> show streaming flash {configuration | statistics}
Displays Flash configuration.

> show streaming ms-smooth {configuration}
View the current Microsoft Smooth configuration.

> show streaming quicktime {configuration | statistics}
Displays QuickTime configuration.

> show streaming real-media {configuration | statistics}
Displays Real-Media configuration.

> show streaming windows-media {configuration | statistics}
Displays Windows-Media configuration and statistics.

> show streaming statistics
Displays client and gateway bandwidth statistics.

For More Information

- "Managing Streaming Media" chapter in *SGOS 6.5.x Administration Guide*

Examples

```
> show streaming configuration
; Streaming Configuration
max-client-bandwidth:    unlimited
max-gateway-bandwidth:   unlimited
multicast address:       224.2.128.0 - 224.2.255.255
multicast port:          32768 - 65535
multicast TTL:           16


> show streaming Adobe-HDS configuration
; Adobe HTTP Dynamic Streaming Configuration
http-handoff: enable
```

> traceroute

Use this command to trace the route from the current host to the specified destination host.

Syntax

```
> traceroute [subcommands]
```

Subcommands

> traceroute *ip_address*
Specifies the IP address of the destination host.

> traceroute *hostname*
Specifies the name of the destination host.

Example

```
SGOS> traceroute 10.25.36.47
Type escape sequence to abort.
Tracing the route to 10.25.36.47
1 10.25.36.47 0 0 0
```

Privileged Mode Commands

Privileged mode provides a set of commands that enable you to view, manage, and change ProxySG settings for features such as log files, authentication, caching, DNS, HTTPS, packet capture filters, and security. You can cannot configure functionality such as SSL Proxy, HTTP compression, and the like.

The prompt changes from a greater than sign (>) to a pound sign (#), acting as an indicator that you are in privileged mode .

Enter privileged mode from standard mode by using the enable command:

```
SGOS> enable
Enable Password:********
SGOS#
```

acquire-utc

Synopsis

Use this command to acquire the Universal Time Coordinates (UTC) from a Network Time Protocol (NTP) server. To manage objects, a ProxySG must know the current UTC time. Your ProxySG comes pre-populated with a list of NTP servers available on the Internet, and attempts to connect to them in the order they appear in the NTP server list on the NTP tab. If the ProxySG cannot access any of the listed NTP servers, the UTC time must be set manually. For instructions on how to set the UTC time manually, refer to "Accessing the ProxySG" in the *SGOS Administration Guide*.

Syntax

```
# acquire-utc
```

The `acquire-utc` command has no parameters or subcommands.

Example

```
SGOS# acquire-utc
 ok
```

bridge

Synopsis

This command clears bridge data.

Syntax

```
# bridge {subcommands]
```

Subcommands

 # bridge clear-statistics *bridge_name*
Clears bridge statistics.

 # bridge clear-fwtable *bridge_name*
Clears bridge forward table.

For More Information

- "Software and Hardware Bridges" in *SGOS 6.5.x Administration Guide*

Example

```
SGOS# bridge clear-statistics testbridge
ok
```

cancel-upload

Synopsis

This command cancels a pending access-log upload. The cancel-upload command allows you to stop repeated upload attempts if the Web server becomes unreachable while an upload is in progress. This command sets log uploading back to idle if the log is waiting to retry the upload. If the log is in the process of uploading, a flag is set to the log. This flag sets the log back to idle if the upload fails.

Syntax

```
# cancel-upload [subcommands]
```

Subcommands

```
# cancel-upload all
```
Cancels upload for all logs.

```
# cancel-upload log  log_name
```
Cancels upload for a specified log.

For More Information

- "Creating Custom Access Log Formats" in *SGOS 6.5.x Administration Guide*

Example

```
SGOS# cancel-upload all
 ok
```

clear-arp

Synopsis

The clear-arp command clears the Address Resolution Protocol (ARP) table. ARP tables are used to correlate an IP address to a physical machine address recognized only in a local area network. ARP provides the protocol rules for providing address conversion between a physical machine address (also known as a Media Access Control or MAC address) and its corresponding IP address, and vice versa.

Syntax

```
# clear-arp
```

The `clear-arp` command has no parameters or subcommands.

Example

```
SGOS# clear-arp
 ok
```

clear-cache

Synopsis

This command clears the byte, dns, or object cache. This can be done at any time. However, keep in mind that if any cache is cleared, performance slows down until the cache is repopulated.

Note: `#clear-cache` with no arguments can also be used to clear the object cache.

Syntax

```
# clear-cache [subcommands]
```

Subcommands

`# clear-cache byte-cache`
Clears the byte cache.

`# clear-cache dns-cache`
Clears the DNS cache.

`# clear-cache object-cache`
Sets all objects in the cache to expired.

Example

```
SGOS# clear-cache byte-cache
 ok
```

clear-errored-connections

Synopsis

This command clears historical errored proxied sessions, errored bypassed connections, and errored ADN inbound connections. To view errored proxied sessions in the Management Console, select **Statistics > Sessions > Errored Sessions > Proxied Sessions**. To view errored bypassed connections in the Management console, select **Statistics > Sessions > Errored Sessions > Bypassed Connections**. To view errored ADN inbound connections in the Management Console, select **Statistics > Active Sessions > ADN Inbound Connections**.

Syntax

```
#clear-errored-connections (subcommand)
```

Subcommands

```
# clear-errored-connections [proxied sessions | bypassed connections |
    adn-inbound connections]
```

Clears the historical proxied sessions, bypassed connections, or ADN inbound connections.

clear-statistics

Synopsis

This command clears the bandwidth-management, persistent, and Windows Media, Real Media, and QuickTime streaming statistics collected by the ProxySG. To view streaming statistics from the CLI, use either the `show streaming {quicktime | real-media | windows-media} statistics` or the `show bandwidth-management statistics [bandwidth_class]` commands. To view streaming statistics from the Management Console, go to either **Statistics > Streaming History > Windows Media/Real Media/Quicktime**, or to **Statistics > Bandwidth Mgmt**.

Syntax

```
# clear-statistics [subcommands]
```

Subcommands

 # clear-statistics authentication [error | realm *realm_name*]
Clears the authentication error statistics.

 # clear-statistics bandwidth-management [class *class_name*]
Clears bandwidth-management statistics, either for all classes at one time or for the bandwidth-management class specified

 # clear-statistics default-services
Clears statistics for default services.

 # clear-statistics epmapper
Clears Endpoint Mapper statistics.

 # clear-statistics export
Removes export statistics. Once this command is run, the next export only includes the data accumulated since the `clear-statistics export` command was run.

 # clear-statistics persistent [*prefix*]
Clears statistics that persist after a reboot. You can clear all persistent statistics, or, since statistics are kept in a naming convention of `group:stat`, you can limit the statistics cleared to a specific group. Common prefixes include HTTP, SSL, and SOCKS.

 # clear-statistics quicktime
Clears QuickTime statistics.

 # clear-statistics real-media
Clears Real Media statistics.

 # clear-statistics windows-media
Clears Windows Media statistics.

Example

```
SGOS# clear-statistics windows-media
 ok
```

configure

Synopsis

The privileged mode subcommand `configure`, enables you to manage the ProxySG appliance features.

Syntax

`# config t`

where `conf` refers to `configure` and `t` refers to `terminal`.

This changes the prompt to `#(config)`.

At this point you are in `configure terminal` mode and can make permanent changes to the device.

`# config network` *`url`*

This command downloads a previously loaded web-accessible script, such as a configuration file, and implements the changes in the script onto the system.

For More Information

- Chapter 3: "Privileged Mode Configure Commands" on page 99

Example

```
# conf n http://1.1.1.1/fconfigure.txt
```

disable

Synopsis

The `disable` command returns you to Standard mode from Privileged mode.

Syntax

```
# disable
```

The `disable` command has no parameters or subcommands.

For More Information

- **> enable** on page 15
- **# exit** on page 54

Example

```
SGOS# disable
SGOS>
```

disk

Synopsis

Use the `disk` command to take a disk offline or to re-initialize a disk.

On a multi-disk ProxySG appliance, after issuing the `disk reinitialize disk_number` command, complete the reinitialization by setting it to empty and copying pre-boot programs, boot programs and starter programs, and system images from the master disk to the re-initialized disk. The master disk is the leftmost valid disk. *Valid* indicates that the disk is online, has been properly initialized, and is not marked as invalid or unusable.

Note: If the current master disk is taken offline, reinitialized or declared invalid or unusable, the leftmost valid disk that has not been reinitialized since restart becomes the master disk. Thus as disks are reinitialized in sequence, a point is reached where no disk can be chosen as the master. At this point, the current master disk is the last disk. If this disk is taken offline, reinitialized, or declared invalid or unusable, the ProxySG appliance is restarted.

Reinitialization is done without rebooting the system, although the system should not proxy traffic during reinitialization. The ProxySG operations, in turn, are not affected, although during the time the disk is being reinitialized, that disk is not available for caching. Note that only the master disk reinitialization might restart the ProxySG.

Syntax

```
# disk {subcommands]
```

Subcommands

`# disk offline disk_number`
Takes the disk specified by *disk_number* off line.

`# disk reinitialize disk_number`
Reinitializes the disk specified by *disk_number.*

`# disk decrease-object-limit [force]`
Decrease the object capacity on all disks so that they will be compatible with releases prior to SGOS 6.2.

WARNING: This command should be executed on an idle system only.

On systems that have had their object store capacity increased with the **disk increase-object-limit** command, you will want to decrease the object limit *before* downgrading to pre-6.2 releases. Note that this command preserves configuration, registry settings, policy, licensing files, and the appliance birth certificate; it does not retain cache contents, access logs, event log, or sysinfo snapshots. If the disk already has the decreased object capacity, the disk will not be modified; the command will be aborted.

The **force** option decreases the object limit without prompting or warning.

This command will not work on a single disk system.

WARNING: If you do not decrease the object store capacity before downgrading to a pre-6.2 image, the disks will be re-initialized after the downgrade and all data and settings will be lost.

`# disk increase-object-limit [force]`
Increase disk object capacity on multi-disk, large-drive systems in order to store more objects on each disk. The increased object capacity is the default for all multi-disk systems that are manufactured with SGOS 6.2; to get this extra capacity on other systems, you have to initiate this command. Note that the disks will be re-initialized in a format that is not compatible with SGOS releases prior to 6.2. After disk re-initialization, the configuration, registry settings, policy, licensing

files, and the appliance birth certificate are preserved; it does not retain cache contents, access logs, event log, and sysinfo snapshots. If the disk already has the increased object capacity, the disk will not be modified; the command will be aborted.

WARNING: This command should be executed on an idle system only.

The **`force`** option increases the object limit without prompting or warning.

This command will not work on a single disk system.

WARNING: Before downgrading to a pre-6.2 release, you must use the `disk decrease-object-limit` command to decrease the object store capacity. If you fail to do this, all data and settings will be lost after the downgrade.

Example

```
SGOS# disk offline 3
 ok
SGOS# disk reinitialize 3
 ok
```

display

See `> display` on page 14 for more information.

enable

Synopsis

Use this command to enter Privileged mode. Privileged mode commands enable you to view and change your configuration settings. A password is always required.

Syntax

```
> enable
```

The `enable` command has no parameters or subcommands.

For More Information

- **`# disable`** on page 49
- **`#(config) security password and hashed_password`** on page 346
- **`#(config) security username`** on page 369

Example

```
SGOS> enable
Enable Password:******
SGOS# conf t
SGOS(config)
```

Where `conf t` is a shortcut to typing `configure terminal`.

exit

Synopsis

Exits from Configuration mode to Privileged mode, from Privileged mode to Standard mode. From Standard mode, the `exit` command closes the CLI session.

Syntax

```
# exit
```

The `exit` command has no parameters or subcommands.

Example

```
SGOS# exit
```

fips-mode

Synopsis

Use this command to enable and disable FIPS mode.

Discussion

When you enter FIPS mode, all previous configurations are destroyed. When you exit FIPS mode, all FIPS configurations are destroyed.

Syntax

```
SG# fips-mode {subcommands}
```

Subcommands

 # fips-mode enable

Enables FIPS mode.

 # fips-mode disable

Disables FIPS mode

Example

```
SGOS# fips-mode enable
```

help

See Accessing Quick Command Line Help on page 11 for information about this command.

hide-advanced

Synopsis

Use this command to disable advanced commands.

Note: You can also use the configure command SGOS#(config) `hide-advanced {all | expand}` to hide commands.

Syntax

`# hide-advanced [subcommands]`

Subcommands

`# hide-advanced all`
Hides all advanced commands.

`# hide-advanced expand`
Disables expanded commands.

For More Information

- `# reveal-advanced` on page 77

Example

```
SGOS# hide-advanced expand
  ok
SGOS# hide-advanced all
  ok
```

inline

Synopsis

Installs lists based on your terminal input.

Discussion

The easiest way to create installable lists, such as forwarding hosts, PAC files, and policy files, among others, is to take an existing file and modify it, or to create the text file on your local system, upload the file to a Web server, and download the file to the ProxySG. As an alternative, you can enter the list directly into the ProxySG through the inline command, either by typing the list line by line or by pasting the contents of the file.

If you choose to create a text file to contain the configuration commands and settings, be sure to assign the file the extension `.txt`. Use a text editor to create this file, noting the following ProxySG configuration file rules:

- Only one command (and any associated parameters) permitted, per line
- Comments must begin with a semicolon (;)
- Comments can begin in any column, however, all characters from the beginning of the comment to the end of the line are considered part of the comment and, therefore, are ignored

Tips:

- When entering input for the inline command, you can correct mistakes on the current line using the backspace key. If you catch a mistake in a line that has already been terminated with the Enter key, you can abort the inline command by typing <Ctrl-C>. If the mistake is caught after you terminate input to the inline command, you must re-enter the entire content.
- The end-of-input marker is an arbitrary string chosen by the you to mark the end of input for the current inline command. The string can be composed of standard characters and numbers, but cannot contain any spaces, punctuation marks, or other symbols.

  Choose a unique end-of-input string that does not match any string of characters in the configuration information. One recommended end-of-input string is `'''` (three single quotes).

Syntax

inline {*subcommands*}

Subcommands

inline accelerated-pac ***eof_marker***
Updates the accelerated pac file with the settings you include between the beginning *eof_marker* and the ending *eof_marker*.

inline authentication-form ***form_name*** *eof_marker*
Install an authentication form from console input

inline authentication-forms *eof_marker*
Install all authentication form from console input

inline banner *eof_marker*
Updates the login banner for the telnet and SSH consoles with the settings you include between the beginning *eof_marker* and the ending *eof_marker*.

 # inline exceptions *eof_marker*
Install exceptions with the settings you include between the beginning *eof_marker* and the ending *eof_marker*.

 # inline forwarding *eof_marker*
Updates the forwarding configuration with the settings you include between the beginning *eof_marker* and the ending *eof_marker*.

 # inline license-key *eof_marker*
Updates the current license key settings with the settings you include between the beginning *eof_marker* and the ending *eof_marker*.

 # inline policy *eof_marker*
Updates the current policy settings—central, local, forward, vpm-cpl, and vpm-xml—with the settings you include between the beginning eof_marker and the ending eof_marker.

 # inline rip-settings *eof_marker*
Updates the current RIP settings with the settings you include between the beginning *eof_marker* and the ending *eof_marker*.

 # inline socks-gateways *eof_marker*
Updates the current SOCKS gateway settings with the settings you include between the beginning *eof_marker* and the ending *eof_marker*.

 # inline static-route-table *eof_marker*
Updates the current static route table settings with the settings you include between the beginning *eof_marker* and the ending *eof_marker*. IP addresses can be IPv4 or IPv6.

 # inline wccp-settings *eof_marker*
Updates the current WCCP settings with the settings you include between the beginning *eof_marker* and the ending *eof_marker*.

For More Information

- **# load** on page 62

Example

```
SGOS# inline wccp eof
wccp enable eof
```

kill

Synopsis

Terminates a CLI session.

Syntax

```
# kill session_number
```

where *session_number* is a valid CLI session number.

Example

```
> show sessions
Sessions:
#   state type              start                          elapsed
   01 IDLE
   02 PRIVL ssh        08 Aug 2006 21:27:51 UTC  23:08:04
   03* NORML ssh     10 Aug 2006 20:35:40 UTC 00:00:15
   ...
> enable
Enable Password:
# kill 3
 ok
```

licensing

Synopsis

Use these commands to request or update licenses.

Syntax

```
# licensing [subcommands]
```

Subcommands

 # licensing request-key [**force**} *user_id password*
Requests the license key from Blue Coat using the BTO user ID and password.

 # licensing update-key [**force**]
Updates the license key from Blue Coat now.

 # licensing register-hardware [**force**] *user_ID password*
Register hardware with Blue Coat.

 # licensing mark-registered
Mark the hardware registered manually.

 # licensing disable-trial
Disable trial period.

 # licensing enable-trial
Enable trial period.

For More Information

- "Licensing" in *SGOS 6.5.x Administration Guide*

Example

```
SGOS# licensing request-key
User ID: admin
Password: *****
...
 ok
```

where ". . ." represents license download-in-progress information.

load

Synopsis

Downloads installable lists or system upgrade images. These installable lists or settings also can be updated using the `inline` command.

Syntax

 # load accelerated-pac
Downloads the current accelerated pac file settings.

 # load authentication-form *form_name*
Downloads the new authentication form.

 # load authentication-forms
Downloads the new authentication forms.

 # load banner
Configure the login banner for the telnet and SSH consoles.

 # load crl *crl_list*
Loads the specified CRL list.

 # load exceptions
Downloads new exceptions.

 # load forwarding
Downloads the current forwarding settings.

 # load keydata [<*passphrase*>]
Loads the keyrings and keylists from the location specified with **keydata-path**.

 # load license-key
Downloads the new license key.

 # load policy {central | forward | local | vpm-cpl | vpm-xml}
Downloads the policy file specified

 # load proxy-client-software
Loads the ProxyClient software to the Client Manager. To use this command, you must have previously defined an upload location using **#(config) sg-client** on page 379. Messages display as the software loads.

 # load rip-settings
Downloads new RIP settings.

 # load socks-gateways
Downloads the current SOCKS gateways settings.

 # load static-route-table
Downloads the current static route table settings.

 # load trust-package
Downloads and installs the trust package from the specified download path. For information on setting the download path and other trust package download settings, see **#(config) security trust-package** on page 367. Note that any manual changes you have made to the `browser-trusted` or `image-validation` CA Certificate Lists (CCLs) or their associated CA certificates will be preserved. The trust package at the specified download path will only be downloaded and installed if signature validation succeeds and if the timestamp on the trust package indicates that it is a newer version than the existing trust packages that have been downloaded to the ProxySG appliance.

 # load upgrade [ignore-warnings]
Downloads the latest system image. The `ignore-warnings` option allows you to force an upgrade even if you receive policy deprecation or disk compatibility warning. Keep the following in mind when using the `ignore-warnings` option:

If you use the `load upgrade ignore-warnings` command to force an upgrade while the system emits deprecation warnings results in a policy load failure; all traffic is allowed or denied according to default policy.

If you use the `load upgrade ignore-warnings` command to force an upgrade while the system emits disk layout incompatibility warnings, the disks will be re-initialized after the downgrade and all data and settings will be lost.

 # load wccp-settings
Downloads the current WCCP settings.

 # load timezone-database
Downloads a new time zone database.

For More Information

- **# inline** on page 58

Example

```
> show download-paths
Policy
  Local:
  Forward:
  VPM-CPL:
  VPM-XML:
  Central:  https://download.bluecoat.com/release/SG3/files/CentralPolicy.txt
    Update when changed: no
    Notify when changed: no
    Polling interval:    1 day
  Accelerated PAC:
  RIP settings:
  Static route table:
  Upgrade image:
    bcserver1.bluecoat.com/builds/ca_make.26649/wdir/8xx.CHK_dbg
  WCCP settings:
  Forwarding settings:
  SOCKS gateway settings:
  License key:
  Exceptions:
  Authentication forms:
>en
  Enable Password
# load upgrade
   Downloading from
"bcserver1.bluecoat.com/builds/ca_make.26649/wdir/8xx.CHK_dbg"
   Downloading new system software (block 2611)
   The new system software has been successfully downloaded.
   Use "restart upgrade" to install the new system software.
```

pcap

Synopsis

The PCAP utility enables you to capture packets of Ethernet frames entering or leaving a ProxySG. Packet capturing allows filtering on various attributes of the frame to limit the amount of data collected. The collected data can then be transferred to the desktop for analysis.

Note: Before using the PCAP utility, consider that packet capturing doubles the amount of processor usage performed in TCP/IP.

To view the captured packets, you must have a tool that can read Packet Sniffer Pro 1.1 files.

Syntax

```
# pcap [subcommands]
```

Subcommands

 # pcap filter on page 65
Specifies filters to use for PCAP.

 # pcap info
Displays the current packet capture information.

 # pcap start on page 67
Starts the capture.

 # pcap stop
Stops the capture.

 # pcap transfer *full_url/filename username password*
Transfers captured data to an FTP site.

For More Information

- *SGOS 6.5.x Administration Guide*

Example 1

Capture transactions among a ProxySG (`10.1.1.1`), a server (`10.2.2.2`), and a client (`10.1.1.2`).

```
SGOS# pcap filter expr “host 10.1.1.1 || host 10.2.2.2 || host 10.1.1.2”
```

Example 2

This example transfers captured packets to the FTP site 10.25.36.47. Note that the username and password are provided.

```
SGOS# pcap transfer ftp://10.25.36.47/path/filename.cap username password
```

If the folders in the path do not exist, they are not created. An error message is generated.

pcap filter

Synopsis

After a filter is set, it remains in effect until it is redefined; the filtering properties are persistent across reboots. However, PCAP stops when a system is rebooted.

Syntax

```
# pcap filter [subcommands]
```

Subcommands

 # pcap filter [direction {in | out | both}]
Specifies capture in the specified direction. If `both` is selected, both incoming and outgoing packets are captured. The default setting is `both`.

 # pcap filter [interface *adapter_number:interface_number* **| all]**
Specifies capture on the specified interface or on all interfaces, such as 0:1. The interface number must be between 0 and 16. The default setting is `all`.

 # pcap filter [expr *filter_expression*]
Specifies capture only when the filter expression matches.

 # pcap filter
No filtering specified (captures all packets in both directions—on all interfaces).

For More Information

- *SGOS 6.5.x Administration Guide*

Example

This example configures packet capturing in both directions, on all interfaces, to or from port 3035:

```
# pcap filter direction both interface all expr “port 3035”
 ok
```

To verify the settings before starting PCAP, enter `pcap info`:

```
SGOS# pcap info
Current state:                    Stopped
Filtering:                             On
Filter:         direction both interface all expr "port 3035"
Packet capture information:
Packets captured:                      0
Bytes captured:                        0
Packets written:                       0
Bytes written:                         0
Coreimage ram used:                    0B
Packets filtered through:              0
```

To start PCAP, enter `pcap start`. Then run `pcap info` to view the results of the packet capture.

```
SGOS# pcap start
 ok
SGOS# pcap info
Current state:                     Capturing
Filtering:                              On
Filter:         direction both interface all expr "port 3035"
Packet capture information:
first count 4294967295 capsize 100000000 trunc 4294967295 coreimage 0
Packets captured:                    2842
Bytes captured:                    237403
Packets written:                     2836
Bytes written:                     316456
Coreimage ram used:                     0B
Packets filtered through:            8147
```

After PCAP is stopped (using the `pcap stop` command), enter `pcap info` to view the results of your PCAP session. You should see results similar to the following:

```
SGOS# pcap info
Current state:                     Stopped
Filtering:                              On
Filter:         direction both interface all expr "port 3035"
Packet capture information:
Packets captured:                       5101
Bytes captured:                         444634
Packets written:                        5101
Bytes written:                          587590
Coreimage ram used:                     0B
Packets filtered through:               10808
```

pcap start

Synopsis

Start packet capture. The `pcap start` options are not persistent across reboots. You must reconfigure them if you reboot the system. The `capsize` and `coreimage` subcommands are used to specify the size of the PCAP file. When no `capsize` or `coreimage` value is specified, the default packet capture file size is 100MB.

Syntax

```
# pcap start [subcommands]
```

Subcommands

`[buffering-method]`

Syntax: `[first | last] {[count <N>]|[capsize <NKB>]}`

The buffering method specifies how captured packets are buffered in memory.

`[count]` and `[capsize]`

The `count` option specifies that the buffer limit is controlled by the number of packets stored in the buffer. The value of count must be between 1 and 1000000.

The `capsize` option specifies the maximum number of bytes stored in the buffer. The `capsize` value is limited to 3% of the available system memory at startup (not to exceed 4GB). This value will differ by appliance model.

Note: The `capsize` *n* option is an approximate command; it captures an approximate number of packets. The actual size of the file written to disk is a little larger than the `capsize` value because of extra packet information such as time-stamps. If no parameters are specified, the default is to capture until the stop subcommand is issued or the maximum limit reached.

`[first]` and `[last]`

The `first` and `last` options affect the buffering behavior when the buffer is full. When `first` is specified, PCAP stops when the buffer limit is exceeded. When `last` is specified, PCAP continues capturing even after the buffer limit has been exceeded. The oldest captured packets are removed from buffer to make space for the newly captured packets: In this way, PCAP captures the last N (or N K bytes of) packets. The saved packets in memory are written to disk when the capture is terminated.

The packet capture file size is limited to 1% of total RAM, which might be reached before n packets have been captured.

Note: The `first` option is a specific command; it captures an exact number of packets. If no parameters are specified, the default is to capture until the stop subcommand is issued or the maximum limit reached.

`[coreimage n]`

Specifies kilobytes of packets kept in a core image. The `coreimage` value is limited to 3% of the available system memory at startup (not to exceed 4GB). This value will differ by appliance model.

[trunc *n*]

The `trunc` *n* parameter collects, at most, *n* bytes of packets from each frame when writing to disk. The range is 1 to 65535.

For More Information

- *SGOS 6.5.x Administration Guide*

Example 1

The following command captures the first 2000 packets that match the filtering expression:

```
# pcap start first count 2000
```

Note that the `first` option configures PCAP to stop capturing after the buffer limit of 2000 packets has been reached. If the `last` option had been specified, PCAP keeps capturing packets even after the buffer limit had been exceeded, until halted by the `pcap stop` command.

Example 2

The following command stops the capturing of packets after approximately three kilobytes of packets have been collected.

```
SGOS# pcap start first capsize 3
```

Example 3

The following command configures the ProxySG appliance to capture 110MB into bluecoat.cap.

```
sgos# pcap start first capsize 110000
```

Example 4

To determine the maximum PCAP file size for your appliance, run the following command:

```
sgos# pcap start first capsize 9999999
Packet capsize must be between 1 and 111184
```

Example 5

If a `capsize` and `coreimage` value are both specified, the maximum of the two values is used for both. For example:

```
sgos# pcap start coreimage 110000 first capsize 105000
```

In this example, 110MB will be captured into bluecoat.cap and into the core image memory.

ping

Synopsis

Use this command to verify that a particular IP address exists and can accept requests. Ping output also tells you the minimum, maximum, and average time it took for the ping test data to reach the other computer and return to the origin.

Syntax

```
> ping {IPv4 address | hostname}
```

Subcommands

> ping *IPv4 address*
Specifies the IPv4 address you want to verify.

> ping *hostname*
Specifies the name of the host you want to verify.

Example

```
SGOS> ping 10.25.36.47
Type escape sequence to abort.
Sending 5, 64-byte ICMP Echos to 10.25.36.47, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5),
round-trip min/avg/max = 0/0/1 ms
Number of duplicate packets received = 0
```

policy

Synopsis

Use this command to configure policy commands.

Note: Configuring the policy command to trace all transactions by default can significantly degrade performance and should only be used in situations where a problem is being diagnosed.

Syntax

```
# policy trace {all | none | proxy-traffic}
```

Use `all` to trace all transactions by default, use `none` to specify no tracing except as specified in policy files, and `proxy-traffic` to trace all proxy transactions by default.

Example

```
policy trace all
  ok
All requests will be traced by default;
Warning: this can significantly degrade performance.
Use 'policy trace none' to restore normal operation
SGOS# policy trace none
  ok
```

register-with-director

Synopsis

The `register-with-director` command is a setup command that automatically registers the ProxySG with a Blue Coat Director, thus enabling that Director to establish a secure administrative session with the. During the registration process, Director can "lock out" all other administrative access to the appliance so that all configuration changes are controlled and initiated by Director.

If your appliance does not have an appliance certificate, you must specify the registration password that is configured on Director.

Syntax

```
# register-with-director dir_ip_address [appliance_name dir_serial_number]
```

Example

```
SGOS# register-with-director 192.0.2.0
Registration Successful
```

remove-sgos7-config

Removes the SGOS 7.x configuration file so that when upgrading from SGOS 6.x to 7.x, the configuration settings for 7.x will be based on the current 6.x configuration.

Syntax

```
# remove-sgos7-config
```

Example

```
SGOS# remove-sgos7-config
Removing SGOS 7.x configuration will permanently delete existing 7.x
configuration from disk.
Continue? (y/n)[n]: y
ok
```

Or if there is no SGOS 7.x configuration found:

```
SGOS# remove-sgos7-config
% No SGOS 7.x configuration is available on this system.
```

For More Information

- **# restore-sgos5-config** on page 75

reset-ui

Synopsis

Restores the Blue Coat Sky user-interface from the system image.

Syntax

```
# reset-ui
```

Example

```
SGOS# reset-ui
Resetting UI to bound system version...
   ok
```

restart

Synopsis

Restarts the system. The restart options determine whether the ProxySG should simply reboot (regular) or reboot using the new image previously downloaded using the `load upgrade` command (upgrade).

Syntax

```
# restart [subcommands]
```

Subcommands

 # restart abrupt

Reboots the system abruptly, according to the version of the ProxySG that is currently installed. Restart abrupt saves a core image. Note that the restart can take several minutes using this option.

 # restart regular

Reboots the version of the ProxySG that is currently installed

 # restart upgrade

Reboots the entire system image and allows you to select the version you want to boot, not limited to the new version on the system.

 # restart upgrade keep-sgos6-config

Reboots the entire system image and preserves the existing SGOS 6.x configuration file. This command only applies when upgrading from SGOS 6.x to 7.x.

For More Information

- **# load** on page 62

Example

```
SGOS# restart upgrade
  ok
SGOS# Read from remote host 203.0.113.0: Connection reset by peer
Connection to 203.0.113.0 closed.
```

restore-sgos5-config

Restores the ProxySG to settings last used with SGOS 5.x. The ProxySG retains the network settings. Note that a reboot is required to complete this command.

Syntax

```
# restore-sgos5-config
```

Example

```
SGOS# restore-sgos5-config
Restoring SGOS 5.x configuration requires a restart to take effect.
The current configuration will be lost and the system will be restarted.
Continue with restoring? (y/n)[n]: y
Restoring configuration ...
```

Or if there is no SGOS 5.x configuration found:

```
SGOS# restore-sgos5-config
%% No SGOS 5.x configuration is available on this system.
```

For More Information

- **# restore-defaults** on page 76

restore-defaults

Synopsis

Restores the ProxySG to the default configuration. When you restore system defaults, the ProxySG's IP address, default gateway, and the DNS server addresses are cleared. In addition, any lists (for example, forwarding or bypass) are cleared. After restoring system defaults, you need to restore the ProxySG's basic network settings and reset any customizations.

Syntax

```
# restore-defaults [subcommands]
```

Subcommands

 # restore-defaults factory-defaults

Reinitializes the ProxySG to the original settings it had when it was shipped from the factory

 # restore-defaults force

Restores the system defaults without confirmation.

If you don't use the `force` command, you are prompted to enter `yes` or `no` before the restoration can proceed.

 # restore-defaults keep-console [force]

Restores defaults except settings required for console access. Using the `keep-console` option retains the settings for all consoles (Telnet-, SSH-, HTTP-, and HTTPS-consoles), whether they are enabled, disabled, or deleted.

If you use the force command, you are not prompted to enter yes or no before restoration can proceed.

For More Information

- "Maintaining the ProxySG" in *SGOS 6.5.x Administration Guide*

Example

```
SGOS# restore-defaults
Restoring defaults requires a restart to take effect.
The current configuration will be lost and the system will be restarted.
Continue with restoring? (y/n)[n]: n
Existing configuration preserved.
```

reveal-advanced

Synopsis

The `reveal-advanced` command allows you to enable all or a subset of the advanced commands available to you when using the CLI. You can also use SGOS#(config) `hide-advanced {all | expand}` to reveal hidden commands.

Syntax

```
# reveal-advanced [subcommands]
```

Subcommands

 # reveal-advanced all
Reveals all advanced commands.

 # reveal-advanced expand
Enables expanded commands.

For More Information

- **# hide-advanced** on page 57

Example

```
SGOS# reveal-advanced all
 ok
```

show

The # show command displays all the show commands available in the standard mode plus the show commands available only in privileged mode and configuration mode. Only show commands available in privileged mode are discussed here. For show commands also available in the standard mode, see **> show** on page 21.

Synopsis

Use this command to display system information.

Syntax

show [*subcommands*]

Subcommands

 # show adn on page 83
Displays ADN configuration.

show archive-configuration
Displays archive configuration settings.

 # show attack-detection on page 84
Displays client attack-detection settings.

 # show configuration on page 86
Displays system configuration.

show connection-forwarding
Displays TCP connection forwarding status and peer IP address list.

 # show content on page 87
Displays content-management commands.

show content-filter {bluecoat | i-filter | intersafe | iwf | local | optenet | proventia | surfcontrol | status | webwasher}
Shows settings for Blue Coat Web Filter or the various third-party content-filtering vendors. You can get information on current content-filtering status by using the # show content-filter status command.

 # show geolocation on page 88
Displays geolocation settings.

show proxy-client
Displays ProxyClient settings.

 # show proxy-services on page 89
Displays information on static and dynamic bypass and proxy-service behavior.

show realms
Displays the status of each realm.

 # show security on page 90
Displays security settings.

 # show ssh-console on page 91
Displays SSH settings.

show session-monitor
Displays the session monitor, which monitors RADIUS accounting messages and maintains a session table based on the information in these messages.

 # show ssl on page 92
Also available in standard mode, the # show ssl command offers more options in privileged mode.

 # show statistics-export
Shows the settings for exporting statistics. This command displays the same information as the #(config statistics-export) view command. See **#(config) statistics-export** on page 417.

 # show system-resource-metrics
Displays system resource statistics.

Examples

```
# show archive-configuration
Archive configuration
  Protocol: FTP
  Host:
  Path:
  Filename:
  Username:
  Password: ************
# show content-filter status
Provider:                               Blue Coat
Status:                                 Ready
Lookup mode:                            Always
Download URL:           https://list.bluecoat.com/bcwf/activity/download/bcwf.db
Download Username:                      BCWF-AUG1511
Automatic download:                     Enabled
Check for updates:                      All day
Category review message:                Enabled
Dynamic Categorization:
  Service:                              Disabled
  Mode:                                 Real-time
  Secure:                               Disabled
  Forward Target:                       <none>
  SOCKS Gateway Target:                 <none>
  Send request info:                    Enabled
  Send malware info:                    Enabled
Download log:
  Blue Coat download at: 2011/09/14 22:50:18 +0000
  Downloading from https://list.bluecoat.com/bcwf/activity/download/bcwf.db
  Requesting differential update
  Update cache entries: 1422
  Update cache version: 312570520
  File has not changed since last download attempt; no download required
Previous download:
  Blue Coat download at: 2011/09/14 20:26:50 +0000
  Downloading from https://list.bluecoat.com/bcwf/activity/download/bcwf.db
  Requesting differential update
  Download size:        57592
  Differential update applied successfully
  Database size:        276557578
  Database date:        Wed, 14 Sep 2011 20:14:41 UTC
  Database expires:     Fri, 14 Oct 2011 20:14:41 UTC
  Database version:     312570500
  Database format:      1.1
Memory Allocation:                      Normal
```

```
CPU Throttle:                           Enabled
# show realms
Local realm:
  No local realm is defined.
RADIUS realm:
  Realm name:                   RADIUS1
  Display name:                 RADIUS1
  Case sensitivity:             enabled
  Primary server host:          10.9.59.210
  Primary server port:          1812
  Primary server secret:        ************
  Alternate server host:
  Alternate server port:        1812
  Alternate server secret:      ************
  Server retry count:           5
  Cache duration:               900
  Virtual URL:
  Server timeout:               5
  Spoof authentication:         none
  One time passwords:           no
LDAP realm(s):
  No LDAP realms are defined.
#show system-resource-metrics
Title Health Monitor Stats
Version 1.1

Overall Health
Current State                : OK
Last Transition              : Thu, 20 Sep 2012 14:50:13 UTC

Health Stats

Stat: CPU Utilization
Current State                : OK
Last Transition              : Thu, 20 Sep 2012 14:49:52 UTC
Current Value                : 1
Unit of Measurement          : percent
Warning Threshold            : 80
Warning Interval             : 120
Critical Threshold           : 95
Critical Interval            : 120
Notification Method          : log

Stat: Memory Utilization
Current State                : OK
Last Transition              : Thu, 20 Sep 2012 14:49:52 UTC
Current Value                : 81
Unit of Measurement          : percent
Warning Threshold            : 90
Warning Interval             : 120
Critical Threshold           : 95
Critical Interval            : 120
Notification Method          : log

Stat: Interface 0:0 Utilization
Current State                : OK
Last Transition              : Thu, 20 Sep 2012 14:49:52 UTC
Current Value                : 0
```

```
Unit of Measurement           : percent
Warning Threshold             : 60
Warning Interval              : 120
Critical Threshold            : 90
Critical Interval             : 120
Notification Method           : log

Stat: Interface 0:1 Utilization
Current State                 : OK
Last Transition               : Thu, 20 Sep 2012 14:49:52 UTC
Current Value                 : 0
Unit of Measurement           : percent
Warning Threshold             : 60
Warning Interval              : 120
Critical Threshold            : 90
Critical Interval             : 120
Notification Method           : log

Stat: Disk 1 Status
Current State                 : OK
Last Transition               : Thu, 20 Sep 2012 14:49:57 UTC
Current Value                 : present
Notification Method           : log, mail, trap

Stat: Motherboard temperature
Current State                 : OK
Last Transition               : Thu, 20 Sep 2012 14:49:52 UTC
Current Value                 : 32.8
Unit of Measurement           : degrees C
High Critical Threshold       : 75.0
High Warning Threshold        : 65.0
Notification Method           : log, mail, trap

Stat: CPU temperature
Current State                 : OK
Last Transition               : Thu, 20 Sep 2012 14:49:52 UTC
Current Value                 : 33.0
Unit of Measurement           : degrees C
High Critical Threshold       : 90.0
High Warning Threshold        : 75.0
Notification Method           : log, mail, trap

Stat: ADN Connection Status
Current State                 : OK
Last Transition               : Thu, 20 Sep 2012 14:49:52 UTC
Current Value                 : Functionality disabled
Notification Method           : log, mail, trap

Stat: ADN Manager Status
Current State                 : OK
Last Transition               : Thu, 20 Sep 2012 14:49:52 UTC
Current Value                 : Not a manager
Notification Method           : log, mail, trap

Stat: Health Check Status
Current State                 : OK
Last Transition               : Thu, 20 Sep 2012 14:50:13 UTC
Current Value                 : OK
Notification Method           : log, mail, trap
```

```
Stat: Cloud Services: Common Policy Expiration
Current State                 : OK
Last Transition               : Thu, 20 Sep 2012 14:49:52 UTC
Current Value                 : Functionality disabled
Unit of Measurement           : days left
Warning Threshold             : 30
Warning Interval              : 0
Critical Threshold            : 0
Critical Interval             : 0
Notification Method           : log

Stat: Cloud Services: Common Policy Error Status
Current State                 : OK
Last Transition               : Thu, 20 Sep 2012 14:49:57 UTC
Current Value                 : Functionality disabled
Unit of Measurement           : hours
Warning Threshold             : 24
Warning Interval              : 0
Critical Threshold            : 48
Critical Interval             : 0
Notification Method           : log
```

show adn

Synopsis

Displays ADN settings and statistics.

Syntax

```
# show adn [subcommands]
```

Subcommands

 # show adn byte-cache
Displays ADN byte-cache settings.

 # show adn routing [**advertise-internet-gateway** | **server-subnets**]
Displays ADN routing settings.

 # show adn tunnel
Displays ADN tunnel configuration.

For More Information

- *SGOS 6.5.x Administration Guide*

Example

```
# show adn
Application Delivery Network Configuration:
  ADN:                      disabled
  Manager port:             3034
  Tunnel port:                3035
  Primary manager:     none
  Backup manager:       none
  External VIP:                    none
  Byte-cache Configuration:
    Max number of peers: 10347
    Max peer memory:         30
Tunnel Configuration:
  proxy-processing http:    disabled
  TCP window size:            65536
  reflect-client-ip :                  use-local-ip
Routing Configuration:
  Internet Gateway:             disabled
  Exempt Server subnet:     10.0.0.0/8
  Exempt Server subnet:     172.16.0.0/16
  Exempt Server subnet:        192.168.0.0/16
```

show attack-detection

Synopsis

Displays client attack-detection settings and client and server statistics.

Syntax

```
# show attack-detection [subcommands]
```

Subcommands

client [**blocked** | **connections** | **statistics**]
Displays client attack-detection settings.

client configuration
Displays attack-detection configuration.

server [**statistics**]
Displays server statistics

For More Information

- *SGOS 6.5.x Administration Guide*

show cachepulse

Synopsis

Displays CachePulse statistics, such as license information, registration status, the download URL for the CachePulse database, results of the last download, and the last successful download.

Syntax

```
# show cachepulse
```

Example

```
# show cachepulse
License Type:            Subscription
Licensed Until:          Thu, 01 Jan 2015 00:00:00 UTC
Service:                 Enabled
Download method:         Direct
Last successful download:
  Time:                  Tue, 30 Jul 2013 17:35:00 UTC
  Downloading from: https://subscription.es.bluecoat.com/cachepulse/latestPolicy
  Version:               20130402
```

show configuration

Synopsis

Displays the current configuration, as different from the default configuration.

Syntax

```
# show configuration [subcommands]
```

Subcommands

 # show configuration
Displays all settings

 # show configuration brief
Displays the configuration without inline expansion.

 # show configuration expanded
Displays the configuration with inline expansion.

 # show configuration noprompts
Displays the configuration without `--More--` prompts.

 # show configuration post-setup
Displays the configuration made after console setup.

 # show configuration versions
Displays the configurations saved for each SGOS version.

Example

Assuming non-default settings of:

- policy = <Proxy> DENY
- IP address of 10.167.42.38

```
# show configuration brief
interface 0:0 ;mode
ip-address 10.167.42.38
exit
# show configuration expanded
interface 0:0 ;mode
ip-address 10.167.42.38
exit
!
inline policy local "end-326998078-inline"
<Proxy>
DENY
end-326998078-inline
```

show content

Synopsis

Displays content-management commands. Note that you must be in enable mode to use the `show content` command.

Syntax

```
# show content [subcommands]
```

Subcommands

 # show content outstanding-requests
Displays the complete list of outstanding asynchronous content revalidation and distribute requests.

 # show content priority [regex *regex* **| url** *url*`]`
Displays the deletion priority value assigned to the *regex* or *url*, respectively

 # show content url *url*
Displays statistics of the specified URL To enter file or directory names that contain spaces, substitute each space with the following escape code: `%20`. If the *url* contains a percent sign, substitute the `%` with `%25`.

To show a CIFS file, the *url* should conform to the following format:

`cifs://server/share/path-to-file`

To show HTTP content, the *url* should use the following format:

`http://host:port/path-to-file`

To show FTP content, the *url* should use the following format:

`ftp://host:port/path-to-file`

To show streaming content, the *url* should use one of the following formats:

`rtsp://host:port/path-to-file`

`mms://host:port/path-to-file`

For More Information

- *SGOS 6.5.x Administration Guide*

show geolocation

Displays geolocation settings related to database download status and countries listed in the database.

Syntax

```
# show geolocation [subcommands]
```

Subcommands

 # show geolocation

Displays the download URL for the geolocation database, and results of the last download and the last successful download. This subcommand produces the same output as the `#(config geolocation) view` command.

 # show geolocation countries

Displays a list of countries defined in the geolocation database (if one has been downloaded). In addition, this subcommand also displays system-defined conditions for country names when geolocation is not enabled or licensed, or if the database is otherwise unavailable. This subcommand produces the same output as the `#(config geolocation) view countries` command.

Example

```
#show geolocation
License Type:           Subscription
Licensed Until:         Thu, 01 Jan 2015 00:00:00 UTC
Service:                Enabled
Download method:        Direct
Last successful download:
  Time:                 Wed, 10 Apr 2013 17:16:54 UTC
  Downloading from:     https://subscription.es.bluecoat.com/geoip/database
  Version:              20130402
```

show proxy-services

Synopsis

Information about proxy services

Syntax

```
# show proxy-services [subcommands]
```

Subcommands

 # show proxy-services
Displays all proxy services configured on the system.

 # show proxy-services dynamic-bypass
Displays dynamic-bypass information.

 # show proxy-services services bypass
Display services containing a bypass action.

 # show proxy-services services intercept
Display services containing an intercept action.

 # show proxy-services services *name*
Display services with name substring match.

 # show proxy-services services *proxy*
Display services using a specific proxy.

 # show proxy-services static-bypass
Displays static-bypass information.

For More Information

- *SGOS 6.5.x Administration Guide*

show security

Synopsis

Displays information about security parameters.

Syntax

```
# show security [subcommands]
```

Subcommands

 # show security
Displays all security settings on the system.

 # show security authentication-errors
Displays all authentication errors.

 # show security authentication-forms
Displays authentication forms configured on the system.

 # show security local-user-list
Displays the local user list configured on the system.

 # show security local-user-list-group
Displays the groups in local user list.

 # show security local-user-list-user
User in local user list

 # show security trust-package
Displays information about the trust package download settings and the status of the latest download.

For More Information

- *SGOS 6.5.x Administration Guide*

Example

```
# show security
Account:
  Username:        "admin"
  Hashed Password: $1$it$24YXwuAGbmvQl7zhaeG5u.
  Hashed Enable Password: $1$U1JZbCl1$itmTNhAwhymF2BNwBnum1/
  Hashed Front Panel PIN: "$1$50KI$KR0RtYxQl02Z26cLy.Pq5."
  Management console display realm name: ""
  Web interface session timeout: 15 minutes
  CLI session timeout: 5 minutes
Access control is disabled
Access control list (source, mask):
Flush credentials on policy update is enabled
Default authenticate.mode: auto
Transparent proxy authentication:
  Method: cookie
  Cookie type: session
  Cookie virtual-url: "www.cfauth.com/"
  IP time-to-live: 15
  Verify IP: yes
  Allow redirects: no
```

show ssh-console

Synopsis

Displays the SSH service details.

Syntax

```
# show ssh-console [subcommands]
```

Subcommands

 # show ssh-console client-key [*username*]
Displays the client key fingerprint for the specified username.

Note: If you upgraded from an older version of the ProxySG, you might not need to enter a username.

 # show ssh-console director-client-key [*key_id*]
Displays all client key fingerprints or the client key fingerprint of the specified key ID.

 # show ssh-console host-public-key [sshv1 | sshv2]
Displays the sshv1 or sshv2 host public key. Both keys are displayed if you do not specify a version.

 # show ssh-console user-list
Displays a list of users with imported RSA client keys.

 # show ssh-console versions-enabled
Displays which SSH version or versions are enabled.

For More Information

- *SGOS 6.5.x Administration Guide*

Example

```
# show ssh-console versions-enabled
SSHv2 is enabled.
```

show ssl

Synopsis

Displays SSL settings.

Syntax

show ssl [*subcommands*]

Subcommands

show ssl ca-certificate ***name***
Displays the CA certificate configuration

show ssl ccl [*list_name***]**
Displays currently configured CA certificate lists or configuration for the specified list_name. This option can also be viewed from standard mode.

show ssl certificate *keyring_id*
Displays the certificate configuration for the specified keyring.

show ssl crl *crl_id*
Displays the SSL certificate Revocation List (CRL) of the specified ID.

show ssl external-certificate *name*
Displays external certificate configuration of the specified name.

show ssl intercept
Displays the SSL intercept configuration.

show ssl keypair {des | des3 | unencrypted} *keyring_id*
Displays the keypair. If you want to view the keypair in an encrypted format, you can optionally specify `des` or `des3` before the keyringID. If you specify either `des` or `des3`, you are prompted for the challenge entered when the keyring was created.

show ssl keyring [*keyring_id*]
Displays all keyrings or the keyring of the specified ID.

show ssl secure-signing-request *keyring_id*
Displays signed certificate signing request for the specified keyring.

show ssl signing-request *keyring_id*
Displays the certificate signing request configuration for the specified keyring.

show ssl ssl-client [*ssl_client***]**
Displays information about all SSL clients or the specified SSL client. This option can also be viewed from standard mode.

show ssl ssl-nego-timeout
Displays the SSL negotiation timeout configuration.

show ssl summary {ca-certificate | crl | external-certificate}
Displays the SSL summary information for CA certificates, CRLs, or external certificates.

For More Information

- *SGOS 6.5.x Administration Guide*

Example

```
# show ssl keyring
KeyringID: configuration-passwords-key
  Is private key showable? yes
  Have CSR? no
  Have certificate? no
KeyringID: default
  Is private key showable? yes
  Have CSR? no
  Have certificate? yes
  Is certificate date range valid? yes
  CA: Blue Coat SG200 Series
  Expiration Date: Mar 02 22:25:32 2016 GMT
  Fingerprint: B2:DE:C4:98:58:18:3C:E3:B3:4A:1C:FC:AB:B5:A4:74
```

static-route

This command has been replaced by **# temporary-route** on page 95.

temporary-route

This command is used to manage temporary route entries. After a reboot these routes are lost.

Syntax

```
# temporary-route [subcommands]
```

Subcommands

 # temporary-route add *destination_address netmask gateway_address*
Adds a temporary route entry.

 # temporary-route delete *destination_address*
Deletes a temporary route entry.

test

This command is used to test subsystems. A `test http get` command to a particular origin server or URL, for example, can verify Layer 3 connectivity and also verify upper layer functionality.

Syntax

```
# test http [subcommands]
```

Subcommands

 # test adn *IP_server_address port*
Tests the ADN connection by connecting to a server. The *IP_server_address* can be either IPv4 or IPv6.

 # test dns {*host_name* | *IP_address*} **[ipv4** | **ipv6]** [*DNS_server_IP*]**[bypass-cache]**
Performs a DNS lookup and displays debugging information that describes the lookup.

Note: If you invoke the *DNS_server_IP* option, the **bypass-cache** option is implied and is not required.

 # test geolocation *IP_address*
Displays the country associated with an IP address. You must have a geolocation database and a valid subscription in order to use the geolocation feature.

 # test http get *url*
Does a test GET of an HTTP object specified by *url*.

 # test service <source-ip> <destination-ip> <port-range> [protocol-type]
Perform a test of proxy services to determine how a spefic request will be handled (bypass/intercept) by the ProxySG Appliance, based on client address and destination address and port.

Example

```
SGOS# test service 192.168.1.5 8.21.6.225 80
Service        : External HTTP
Proxy Type     : http
Listener Match : All -> Transparent (80)
Action         : intercept

SGOS# test http get http://www.google.com
Type escape sequence to abort.
Executing HTTP get test
* HTTP request header sent:
GET http://www.google.com/ HTTP/1.0
Host: www.google.com
User-Agent: HTTP_TEST_CLIENT
* HTTP response header recv'd:
HTTP/1.1 200 OK
Connection: close
Date: Tue, 15 Jul 2003 22:42:12 GMT
Cache-control: private
Content-Type: text/html
Content-length: 2691
Set-Cookie:
PREF=ID=500ccde1707c20ac:TM=1058308932:LM=1058308932:S=du3WuiW7FC_lJ
Rgn; expires=Sun, 17-Jan-2038 19:14:07 GMT; path=/; domain=.google.com
Measured throughput rate is 66.72 Kbytes/sec
HTTP get test passed
```

traceroute

Use this command to trace the route to a destination. The `traceroute` command can be helpful in determining where a problem might lie between two points in a network. Use `traceroute` to trace the network path from a ProxySG back to a client or to a specific origin Web server.

Note that you can also use the trace route command from your client station (if supported) to trace the network path between the client, a ProxySG, and a Web server. Microsoft operating systems generally support the trace route command from a DOS prompt. The syntax from a Microsoft-based client is: `tracert` [*ip* | *hostname*].

Syntax

```
# traceroute [subcommands]
```

Subcommands

 # traceroute *IP_address*
Indicates the IP address of the client or origin server.

 # traceroute *hostname*
Indicates the hostname of the origin server.

Example

```
SGOS# traceroute 10.25.36.47
Type escape sequence to abort.
Executing HTTP get test
HTTP response code: HTTP/1.0 503 Service Unavailable
Throughput rate is non-deterministic
HTTP get test passed
10.25.36.47# traceroute 10.25.36.47

Type escape sequence to abort.
Tracing the route to 10.25.36.47
1 10.25.36.47 212 0 0 0
```

upload

Uploads the current access log or running configuration.

Syntax

```
# upload {subcommands}
```

Subcommands

 # upload access-log all
Uploads all access logs to a configured host.

 # upload access-log log *log_name*
Uploads a specified access log to a configured host.

 # upload configuration
Uploads running configuration to a configured host.

Example

```
SGOS# upload configuration
 ok
```

Chapter 3: Privileged Mode Configure Commands

This chapter describes and provides examples for privileged mode configure command, which allows you to configure the ProxySG appliance settings.

Configure Mode Commands

The `configure` command, available only in enabled mode, allows you to configure the Blue Coat ProxySG settings from your current terminal session (`configure terminal`), or by loading a text file of configuration settings from the network (`configure network`).

The prompt changes from a pound sign (#) to a #(config) prompt, acting as an indicator that you are in configuration mode .

Enter configuration mode from privileged mode by using the configure command:

```
SGOS# conf t
SGOS#(config)
```

No password is required to enter configure mode.

#(config) accelerated-pac

Synopsis

Specify the location of the PAC file on a Web server.

Discussion

Typically, the Proxy Auto-Configuration (PAC) file is located on a Web server, and client browsers are served the PAC file from the Web server. Alternatively, you can place the PAC file on the ProxySG, and have client browsers load the file directly from the proxy appliance. This feature accelerates the serving of the PAC file. Use the **accelerated-pac path** command to specify the location of the PAC file on the Web server, and then use the **load accelerated-pac** command to load the PAC file on the ProxySG.

After you have configured the ProxySG to use an accelerated PAC file, you must also configure client browsers with the proxy configuration URL (that is, the URL of the PAC file on the ProxySG). For example, if the PAC file is named `accelerated_pac_base.pac`, specify the following URL for automatic proxy configuration:

```
https://SG_IP_Address:8082/accelerated_pac_base.pac
```

As an alternative to port 8082, you can specify the port that is being intercepted for the explicit HTTP proxy service. For example, if port 8080 is being intercepted, you can specify:

```
http://SG_IP_Address:8080/accelerated_pac_base.pac
```

You might want to use this alternative to avoid overloading the management port with too many client connections while client browsers are retrieving the PAC file.

Syntax

`#(config) accelerated-pac no path`
Clears the network path to download PAC file.

`#(config) accelerated-pac path` *`url`*
Specifies the location on the Web server from which the PAC file should be downloaded.

For More Information

- **# inline** on page 58
- **# load** on page 62
- *SGOS Administration Guide*

Example

```
#(config) accelerated-pac path http://www.comp.com/pac/accelerated_pac_base.pac
#(config) load accelerated-pac
```

#(config) access-log

Synopsis

The ProxySG can maintain an access log for each HTTP request made. The access log can be stored in one of three formats, which can be read by a variety of reporting utilities.

Syntax

```
#(config) access-log
```

This changes the prompt to:

```
#(config access-log)
```

Subcommands

`#(config access-log)` **`create log`** *`log_name`*
Creates an access log.

`#(config access-log)` **`create format`** *`format_name`*
Creates an access log format.

`#(config access-log)` **`cancel-upload all`**
Cancels upload for all logs.

`#(config access-log)` **`cancel-upload log`** *`log_name`*
Cancels upload for a log

`#(config access-log)` **`default-logging`** {**`cifs`** | **`epmapper`** | **`ftp`** | **`http`** | **`https-forward-proxy`** | **`https-reverse-proxy`** | **`im`** | **`mapi`** | **`mms`** | **`p2p`** | **`rtsp`** | **`socks`** | **`ssl`** | **`tcp-tunnel`** | **`telnet`**} *`log_name`*
Sets the default log for the specified protocol.

`#(config access-log)` **`delete log`** *`log_name`*
Deletes an access log.

`#(config access-log)` **`delete format`** *`format_name`*
Deletes an access log format.

`#(config access-log)` **`disable`**
Disables access logging.

`#(config access-log)` **`early-upload`** *`megabytes`*
Sets the log size in megabytes that triggers an early upload.

`#(config access-log)` **`edit log`** *`log_name`*—changes the prompt (see **`#(config log log_name)`** on page 104)

`#(config access-log)` **`edit format`** *`format_name`*—changes the prompt (see **`#(config format format_name)`** on page 108)

`#(config access-log)` **`enable`**
Enables access logging.

`#(config access-log)` **`exit`**
Exits `#(config access-log)` mode and returns to `#(config)` mode.

`#(config access-log)` **`max-log-size`** *`megabytes`*
Sets the maximum size in megabytes that logs can reach.

`#(config access-log) no default-logging {cifs | epmapper | ftp | http | https-forward-proxy | https-reverse-proxy | im | mapi | mms | p2p | rtsp | socks | ssl | tcp-tunnel | telnet}`
Disables default logging for the specified protocol.

`#(config access-log) overflow-policy delete`
Deletes the oldest log entries (up to the entire log).

`#(config access-log) overflow-policy stop`
Stops access logging until logs are uploaded.

`#(config access-log) upload all`
Uploads all logs.

`#(config access-log) upload log log_name`
Uploads a log.

`#(config access-log) view`
Shows access logging settings.

`#(config access-log) view {log {brief | log_name}}`
Shows the entire access log configuration, a brief version of the access log configuration, or the configuration for a specific access log.

`#(config access-log) view {format {brief | format_name}}`
Shows the entire log format configuration, a brief version of the log format configuration, or the configuration for a specific log format.

`#(config access-log) view {statistics {log_name}}`
Shows access log statistics for all logs or for the specified log.

`#(config access-log) view default-logging`
Shows the access log default policy

Example

```
SGOS#(config) access-log
SGOS#(config access-log) create log test
 ok
SGOS#(config access-log) max-log-size 1028
 ok
SGOS#(config access-log) overflow-policy delete
 ok
```

View the results. (This is a partial output.)

```
SGOS#(config access-log) view log
Settings:
Log name: main
Format name: main
Description:
Logs uploaded using FTP client
Logs upload as gzip file
Wait 60 seconds between server connection attempts
FTP client:
Filename format: SG_%f_%l%m%d%H%M%S.log
Filename uses utc time
Use PASV: yes
Use secure connections: no
Primary host site:
Host:
Port: 21
Path:
Username:
```

```
Password: ************
Alternate host site:
Host:
Port: 21
Path:
```

#(config log *log_name)*

Synopsis

Use these commands to edit an access log.

Syntax

```
#(config) access-log
```

This changes the prompt to:

```
#(config access-log)
#(config access-log) edit log log_name
```

This changes the prompt to:

```
#(config log log_name)
```

Subcommands

`#(config log log_name) bandwidth-class bwm_class_name`
Specifies a bandwidth-management class for managing the bandwidth of this log.In order to bandwidth-manage this log, bandwidth management must be enabled. Bandwidth management is enabled by default.

Note: You must also create a bandwidth class for this access log (in bandwidth-management mode) before you can select it here. See **`#(config) bandwidth-management`** on page 135 for more information

---

`#(config log log_name) client-type bluecoat`
Uploads log using the Blue Coat Reporter client.

`#(config log log_name) client-type custom`
Uploads log using the custom client.

`#(config log log_name) client-type ftp`
Uploads log using the FTP client.

`#(config log log_name) client-type http`
Uploads log using the HTTP client.

`#(config log log_name) client-type none`
Disables uploads for this log

`#(config log log_name) commands cancel-upload`
Disables uploads for this log.

`#(config log log_name) commands close-connection`
Closes a manually opened connection to the remote server.

`#(config log log_name) commands delete-logs`
Permanently deletes all access logs on the ProxySG.

`#(config log log_name) commands open-connection`
Manually opens a connection to the remote server.

`#(config log log_name) commands rotate-remote-log`
Switches to a new remote log file.

`#(config log log_name) commands send-keep-alive`
Sends a keep-alive log packet to the remote server.

`#(config log log_name)` **`commands test-upload`**
Tests the upload configuration by uploading a verification file.

`#(config log log_name)` **`commands upload-now`**
Uploads access log now.

`#(config log log_name)` **`connect-wait-time`** *`seconds`*
Sets time to wait between server connect attempts.

`#(config log log_name)` **`continuous-upload`** *`seconds`*

`#(config log log_name)` **`continuous-upload enable`**
Uploads access log continuously to remote server.

`#(config log log_name)` **`continuous-upload keep-alive`** *`seconds`*
Sets the interval between keep-alive log packets

`#(config log log_name)` **`continuous-upload lag-time`** *`seconds`*
Sets the maximum time between log packets (text upload only).

`#(config log log_name)` **`continuous-upload rotate-remote`** {**`daily`** *`rotation_hour (0-23)`* | **`hourly`** *`hours`* [*`minutes`*]}
Specifies when to switch to new remote log file.

`#(config log log_name)` **`custom-client alternate`** *`hostIP-address`* [*`port`*]
Configures the alternate custom server address. The *`hostIP-address`* must be defined as an IPv4 address.

`#(config log log_name)` **`custom-client no`** {**`alternate | primary`**}
Deletes the alternate or primary custom host site.

`#(config log log_name)` **`custom-client primary`** *`hostIP-address`* [*`port`*]
Configures the primary custom server address. The *`hostIP-address`* must be defined as an IPv4 address.

`#(config log log_name)` **`custom-client secure`** {**`no | yes`**}
Selects whether to use secure connections (SSL). The default is set to `no;` in other words, custom-client by default is in no-FIPS mode;

`#(config log log_name)` **`description`** *`description`*
Sets the log description.

`#(config log log_name)` **`early-upload`** *`megabytes`*
Sets log size in megabytes that triggers an early upload.

`#(config log log_name)` **`encryption certificate`** *`certificate_name`*
Specifies access-log encryption settings.

`#(config log log_name)` **`exit`**
Exits `#(config log log_name)` mode and returns to `#(config access-log)` mode.

`#(config log log_name)` **`format-name`** *`format_name`*
Sets the log format.

`#(config log log_name)` **`ftp-client alternate`** {**`encrypted-password`** *`encrypted_password`* | **`host`** *`hostname`* [*`port`*] | **`password`** *`password`* | **`path`** *`path`* | **`username`** *`username`*}
Configures the alternate FTP host site. The *`hostname`* can be defined as an IPv4 or IPv6 address, or a domain name that resolves to an IPv4 or IPv6 address.

`#(config log log_name)` **`ftp-client filename`** *`format`*
Configures the remote filename format

`#(config log log_name)` **`ftp-client no`** {**`alternate | filename | primary`**}
Deletes the remote filename format or the alternate or primary host parameters.

`#(config log log_name)` **`ftp-client pasv`** {**`no | yes`**}
Sets whether PASV or PORT command is sent.

`#(config log` *log_name*`)` **`ftp-client primary`** {**`encrypted-password`** *encrypted_password* | **`host`** *hostname* [*port*] | **`password`** *password* | **`path`** *path* | **`username`** *username*}
Configures the primary FTP host site. The *hostname* can be defined as an IPv4 or IPv6 address, or a domain name that resolves to an IPv4 or IPv6 address.

`#(config log` *log_name*`)` **`ftp-client secure`** {**`no`** | **`yes`**}
Selects whether to use secure connections (FTPS). The default is `no`. If `yes`, the *hostname* must match the hostname in the certificate presented by the server.

`#(config log` *log_name*`)` **`ftp-client time-format`** {**`local`** | **`utc`**}
Selects the time format to use within upload filename.

`#(config log` *log_name*`)` **`http-client alternate`** {**`encrypted-password`** *encrypted_password* | **`host`** *hostname* [*port*] | **`password`** *password* | **`path`** *path* | **`username`** *username*}
Configures the alternate HTTP host site. The *hostname* can be defined as an IPv4 or IPv6 address, or a domain name that resolves to an IPv4 or IPv6 address.

`#(config log` *log_name*`)` **`http-client filename`** *format*
Configures the remote filename format.

`#(config log` *log_name*`)` **`http-client no`** {**`alternate`** | **`filename`** | **`primary`**}
Deletes the remote filename format or the alternate or primary host parameters.

`#(config log` *log_name*`)` **`http-client primary`** {**`encrypted-password`** *encrypted_password* | **`host`** *hostname* [*port*] | **`password`** *password* | **`path`** *path* | **`username`** *username*}
Configures the primary HTTP host site. The *hostname* can be defined as an IPv4 or IPv6 address, or a domain name that resolves to an IPv4 or IPv6 address.

`#(config log` *log_name*`)` **`http-client secure`** {**`no`** | **`yes`**}
Selects whether to use secure connections (HTTPS). The default is no. If yes, the *hostname* must match the hostname in the certificate presented by the server

`#(config log` *log_name*`)` **`http-client time-format`** {**`local`** | **`utc`**}
Selects the time format to use within upload filename.

`#(config log` *log_name*`)` **`no`** {**`encryption`** | **`bandwidth-class`** | **`signing`**}
Disables access-log encryption, bandwidth management, or digital signing for this log.

`#(config log` *log_name*`)` **`periodic-upload enable`**
Uploads access log daily/hourly to remote server.

`#(config log` *log_name*`)` **`periodic-upload upload-interval`** {**`daily`** *upload_hour (0-23)* | **`hourly`** *hours* [*minutes*]}
Specifies access log upload interval.

`#(config log` *log_name*`)` **`remote-size`** *megabytes*
Sets maximum size in MB of remote log files.

`#(config log` *log_name*`)` **`signing keyring`** *keyring_id*
Specifies the keyring to be used for digital signatures.

`#(config log` *log_name*`)` **`upload-type`** {**`gzip`** | **`text`**}
Sets upload file type (gzip or text).

`#(config log` *log_name*`)` **`view`**
Shows log settings.

For More Information

- **`#(config) access-log`** on page 101

Example

```
SGOS#(config) access-log
SGOS#(config access-log) edit log testlog
SGOS#(config log testlog) upload-type gzip
 ok
SGOS#(config log testlog) exit
SGOS#(config access-log) exit
SGOS#(config)
```

#(config format *format_name)*

Synopsis

Use these commands to edit an access log format.

Syntax

```
#(config) access-log
```

This changes the prompt to:

```
#(config access-log) edit format format_name
```

This changes the prompt to:

```
#(config format format_name)
```

Subcommands

`#(config format format_name) exit`
Exits `#(config format format_name)` mode and returns to `#(config access-log)` mode.

`#(config format format_name) multi-valued-header-policy log-all-headers`
Sets multi-valued header policy to log all headers.

`#(config format format_name) multi-valued-header-policy log-first-header`
Sets multi-valued header policy to log the first header.

`#(config format format_name) multi-valued-header-policy log-last-header`
Sets multi-valued header policy to log the last header.

`#(config format format_name) type custom format_string`
Specifies custom logging format.

`#(config format format_name) type elff format_string`
Specifies W3C extended log file format.

`#(config format format_name) view`
Shows the format settings.

For More Information

- **`#(config) access-log`** on page 101

Example

```
SGOS#(config) access-log
SGOS#(config access-log) edit format testformat
SGOS#(config format testformat) multi-valued-header-policy log-all-headers
 ok
SGOS#(config format testformat) exit
SGOS#(config access-log) exit
SGOS#(config)
```

#(config) adn

Synopsis

ADN optimization allows you to reduce the amount of tunneled TCP traffic across a WAN by means of an overlay network called an Application Delivery Network, or ADN. ProxySG devices that participate in the ADN utilize byte caching technology, which replaces large chunks of repeated data with small tokens representing that data. ProxySG devices in the ADN also use gzip compression to further reduce the amount of data flowing over the WAN.

Syntax

```
SGOS#(config) adn
```

The prompt changes to

```
SGOS#(config adn)
```

Subcommands

`SGOS#(config adn) byte-cache`
Configures byte caching parameters. The prompt changes to `SGOS#(config adn byte-cache)`

`SGOS#(config adn byte-cache) exit`
Exits the `SGOS#(config adn byte-cache)` submode and returns to `SGOS#(config adn)` mode.

`SGOS#(config adn byte-cache) adaptive-compression {enable | disable}`
Enables or disables adaptive compression. When adaptive compression is enabled, the ProxySG determines whether to increase or decrease the compression level based on CPU usage. When extra CPU is available, it will adapt compression to use these additional resources, resulting in higher CPU usage.

`SGOS#(config adn byte-cache) delete-peer peer-id [force]`
Deletes the specified ADN peer. If the peer has an established dictionary or a dictionary that is manually sized, you will be prompted to confirm that you want to proceed with the deletion. The force argument allows you to delete a peer without confirmation.

`SGOS#(config adn byte-cache) max-disk-usage percentage`
Sets the maximum percentage of disk space that can be used for byte caching. When this setting is changed, an immediate resizing is done. If the statistics have changed since the last resizing, the recommended dictionary sizes and the rankings for each peer might change. However, if there has been no traffic (and it is still the same day), or if the changes balance out, there might be no change to either the recommended dictionary sizes or the rankings.

`SGOS#(config adn byte-cache) peer-size peer-id {size_in_megabytes | auto | none}`
Manually sets the amount of memory used to keep track of the byte-cache hash table or disables compression to this peer. Generally, the dynamic settings are acceptable; you do not need to change the dictionary size. Only if you determine that the algorithm performance does not guarantee the right dictionary size for a specific peer should you manually set the dictionary size.

`SGOS#(config adn byte-cache) view`
Views the current configuration of the byte caching parameters.

`SGOS#(config adn) {enable | disable}`
Enables or disables the ADN optimization network.

`SGOS#(config adn) exit`
Exits the `SGOS#(config adn)` submode and returns to `SGOS#(config)` mode.

`SGOS#(config adn) load-balancing`
Configures load-balancing parameters. The prompt changes to `SGOS#(config adn load-balancing)`.

`SGOS#(config adn load-balancing) {enable | disable}`
Enables or disables load-balancing functionality.

`SGOS#(config adn load-balancing) exit`
Exits the submode and returns to `SGOS#(config adn)` mode.

`SGOS#(config adn load-balancing) external-vip IP_address`
Sets the external VIP. The *IP_address* can be either IPv4 or IPv6, but must be reachable by all branch peers.The same VIP must be configured on each ProxySG in the cluster, and the VIP must exist on an external load balancing device. The external VIP is used in explicit external load balancing.

`SGOS#(config adn load-balancing) group group_name`
Sets the group name for an ADN group. Groups are used in transparent load balancing.

`SGOS#(config adn load-balancing) load-balance-only {enable | disable}`
Specifies whether the node can take participate in load balancing (`disable`) or if it acts as a load balancer only (`enable`).

`SGOS#(config adn load-balancing) no {external-vip | group}`
Removes the external VIP or group name.

`SGOS#(config adn load-balancing) view`
Views the load-balancing configuration.

`SGOS#(config adn) manager`
Configures manager parameters. The prompt changes to `SGOS#(config adn manager).`

`SGOS#(config adn manager) approved-peers`
Configures approved-peers. The prompt changes to `SGOS#(config adn approved-peers).`

`SGOS#(config adn approved-peers) add peer-serial-number`

`SGOS#(config adn approved-peers) exit`
Exits the `SGOS#(config adn approved-peers)` submode and returns to `SGOS#(config adn manager)` mode.

`SGOS#(config adn approved-peers) remove`
Removes the peer(s) from the approved peers list.

`SGOS#(config adn approved-peers) view`
Views the list of approved devices and connections, as well as the device ID of the ADN manager and backup manager.

`SGOS#(config adn manager) backup-manager {IP_address [device_id] | self | none}`
Defines the backup ADN manager; the *IP_address* can be IPv4 or IPv6. While optional, defining a backup ADN manager is highly recommended. If the primary ADN manager goes offline for any reason, routing updates are no longer available which prevent nodes from learning when other nodes enter and leave the network. Existing route information is still retained by the peers, however.

`SGOS#(config adn manager) exit`
Exits the `SGOS#(config adn manager)` submode and returns to `SGOS#(config adn)` mode.

`SGOS#(config adn manager) open-adn {enable | disable}`
`Enables or disables Open-ADN mode.`

`SGOS#(config adn manager) pending-peers`
Configures pending peers. The prompt changes to `SGOS#(config adn pending-peers)`

`SGOS#(config adn pending-peers) {accept | reject} {device-id | all}`
Allows or denies a specific peer or all peers that want to join a network.

`SGOS#(config adn pending-peers)` **{enable | disable}**
Enables or disables the pending-peers functionality.

`SGOS#(config adn pending-peers)` **exit**
Exits the `SGOS#(config adn pending-peers)` submode and returns to `SGOS#(config adn manager)` mode.

`SGOS#(config adn pending-peers)` **view**
Views the list of pending devices and connections.

`SGOS#(config adn manager)` **port** *port_number*
Sets the port number for the primary and backup ADN managers. All ProxySG devices in the ADN must use the same manager port number. The default is port 3034; it should not be changed.

`SGOS#(config adn manager)` **primary-manager** {*IP_address* [*device_id*] | *self* | *none*}
Defines the primary ADN manager; the *IP_address* can be IPv4 or IPv6. The responsibility of the ADN manager is to keep up to date the routing information from each ProxySG node on the WAN optimization network and to broadcast that information to all the peers.

`SGOS#(config adn manager)` **secure-port** *port_number*

`SGOS#(config adn manager)` **view**
Views the adn manager configuration.

`SGOS#(config adn)` **routing**
Configures routing information. The prompt changes to `SGOS#(config adn routing)`.

`SGOS#(config adn routing)` **advertise-internet-gateway**
Enters `advertise-internet-gateway` mode to enable the ProxySG as an Internet gateway. Changes the prompt to `SGOS#(config adn advertise-internet-gateway).`

`SGOS#(config adn routing advertise-internet-gateway)` **{disable | enable}**
Enables or disables the ability for this peer to be used as an Internet gateway.

`SGOS#(config adn routing advertise-internet-gateway)` **exempt-subnets** {**add** {*subnet prefix*[*/prefix_length*]} **clear-all** | **remove** {*subnet prefix*[*/prefix_length*]} | **view**}
Manages subnets t that must not be routed to Internet gateway(s). The subnets can be IPv4, IPv6, or a combination. The *subnet prefix* can be in either IPv4 or IPv6 format.

`SGOS#(config adn routing advertise-internet-gateway)` **exit**
Leaves the `advertise-internet-gateway` submode and returns to the routing submode.

`SGOS#(config adn routing advertise-internet-gateway)` **view**
Displays the advertise-internet-gateway parameters.

`SGOS#(config adn routing)` **prefer-transparent** {**enable** | **disable**}
Forces peers to always use advertised routes or to allows them to use transparent routes if they are available.

`SGOS#(config adn routing)` **exit**
Exits the `SGOS#(config adn routing)` submode and returns to `SGOS#(config adn)` mode.

`SGOS#(config adn routing)` **server-subnets**
Configures server-subnets that will be advertised to other peers on the WAN optimization network. The server subnets can be IPv4, IPv6, or a combination. The prompt changes to `SGOS#(config adn routing server-subnets).`

`SGOS#(config adn routing server-subnets)` **add** *subnet prefix*[*/prefix length*]
Adds a subnet with the specified prefix and, optionally, the prefix length, to the ProxySG routes that it sends to the ADN manager. The *subnet prefix* can be in either IPv4 or IPv6 format.

`SGOS#(config adn routing server-subnets)` **clear-all**
Deletes all subnets listed on the system.

`SGOS#(config adn routing server-subnets)` **`remove`** *`subnet_prefix`*`[/prefix length]`
Removes a subnet with the specified prefix and, optionally, the prefix length, to the ProxySG routes that it sends to the ADN manager. The *`subnet_prefix`* can be in either IPv4 or IPv6 format.

`SGOS#(config adn routing server-subnets)` **`exit`**
Exits the `SGOS#(config adn routing server-subnets)` submode and returns to `SGOS#(config adn routing)` submode.

`SGOS#(config adn routing server-subnets)` **`view`**
Views the current configuration of the server subnets.

`SGOS#(config adn routing)` **`view`**
Views the current parameters of the routing configuration.

`SGOS#(config adn)` **`security`**
Configures authorization parameters. Changes the prompt to `SGOS#(config adn security)`.

`SGOS#(config adn security)` **`authorization {enable | disable}`**
Enables connection authorization.

`SGOS#(config adn security)` **`exit`**
Leaves the security submode. Returns to `(config adn)` mode.

`SGOS#(config adn security)` **`manager-listening-mode {plain-only | plain-read-only | secure-only| both}`**
Configure manager listening mode. Both refers to `plain-only` or `secure-only`.

`SGOS#(config adn security)` **`no ssl-device-profile`**
Clears the SSL device profile name.

`SGOS#(config adn security)` **`secure-outbound {none | secure-proxies | all}`**
Configure outbound connection encryption, where `none` indicates the encryption is disabled, `secure-proxies` enables encryption on secure proxy (that is, HTTPS or SSL) traffic, and `all` indicates that encryption is enabled on all outbound connections.

`SGOS#(config adn security)` **`tunnel-listening-mode {plain-only | secure-only| both}`**
Starts the specified tunnel listening mode.

`SGOS#(config adn security)` **`view`**
View security configuration.

`SGOS#(config adn)` **`tunnel`**
Configures parameters for tunnel connections. Tunnel connections are established between ADN peers in order to carry optimized traffic over the WAN. Changes the prompt to `SGOS#(config adn tunnel).`

`SGOS#(config adn tunnel)` **`connect-transparent {enable [fast|regular]| disable}`**
Control outbound ADN transparent tunnel initiation. Use the **`regular`** option when the concentrator is running SGOS 5.5 and the branch peer is running SGOS 6.5.x, 6.4.x, 6.3.x, 6.2.2, or 6.1.4.

Note: The **`fast|regular`** options were introduced in 6.2.2.1 and 6.1.4.1.

`SGOS#(config adn tunnel)` **`exit`**
Exits the `SGOS#(config adn tunnel)` submode and returns to `SGOS#(config adn)` mode.

`SGOS#(config adn tunnel)` **`last-peer-detection {enable | disable}`**
Allows traffic to be optimized across the entire data path of a transparent ADN deployment: from the branch office, through one or more intermediate concentrators, all the way to the main data center.

`SGOS#(config adn tunnel)` **`port`** *`port_number`*
Sets the port number for the client or data port used by ADN tunnel connections. Each ADN node

has a TCP listener on this port in order to receive tunnel connections. The default is port 3035; it should not be changed.

`SGOS#(config adn tunnel) preferred-ip-addresses`
Configure a list of preferred tunnel or control IP addresses. By default, the list is empty; this means that all IP addresses configured on the ProxySG are eligible to be used for inbound ADN control connections and explicit tunnel connections. The IP addresses that are not in the preferred list will not be advertised for use in tunnel and control connections. Note that this list indicates a *preference* only; if the concentrator gets an inbound ADN connection on an IP address that is not in the preferred list, that connection is still accepted.

In an open, unmanaged transparent ADN deployment, the concentrator looks at the list of preferred IP addresses and determines which IP address to send to the branch peer by following the guidelines below:

1. The concentrator's first choice is to use a preferred IP address of the same address family as the source address on the interface that the connection came on.
2. If that's not possible, it uses a preferred IP address of the same address family as the source address, on an interface that is different from the interface that the connection came on.
3. If the concentrator can't use an IP from the same address family, the concentrator uses a preferred IP address of a different address family on the interface that the connection came on.
4. If the same interface isn't possible, it uses a preferred IP address of a different address family, on an interface that is different from the interface that the connection came on.
5. If none of the above are applicable, the concentrator uses the first data IP address in the preferred IP list.

   Note: If there isn't a preferred list, the concentrator selects the first IP configured on the incoming tunnel connection interface.

`SGOS#(config adn tunnel preferred-ip-addresses) add IP address`
Add an IP address to the preferred list. The *`IP address`* can be IPv4 or IPv6. This list is communicated to ADN peers so that they can form explicit tunnels and control connections.

`SGOS#(config adn tunnel preferred-ip-addresses) clear-all`
Remove all IP addresses from the preferred list. When the list is empty, all IP addresses configured on the ProxySG are available for tunnel and control connections.

`SGOS#(config adn tunnel preferred-ip-addresses) remove IP address`
Remove an IP address from the preferred list. This IP address will no longer be preferred for tunnel and control connections. Existing control/tunnel connections using a deleted IP address will not be effected; only new connections will use the new configuration.

`SGOS#(config adn tunnel preferred-ip-addresses) view`
View the list of preferred IP addresses.

`SGOS#(config adn tunnel) preserve-dest-port {enable | disable}`
Preserve destination port on outbound connections

`SGOS#(config adn tunnel) proxy-processing http {enable | disable}`
Enables HTTP handoff. This option should be used with care as both byte caching and object caching require significant resources. Be sure that your ProxySG devices are sized correctly if you intend to use this option.

Note: The proxy processing feature has been deprecated. Since proxy processing will be completely removed from an SGOS release in the near future, Blue Coat recommends that you discontinue using this feature and deploy a separate secure web gateway to handle proxy processing.

`SGOS#(config adn tunnel) reflect-client-ip {allow | deny | use-local-ip}`
This CLI command is hidden starting in SGOS 6.2, but it is available for backward compatibility purposes.

Configures the Concentrator peer to follow (`allow`), reject (`deny`), or ignore (`use-local-ip`) the

Branch peer reflect-client-ip settings. When `allow` is specified, both ProxySG and ProxyClient Branch peers will be set to `allow`. When `deny` is specified, ProxySG Branch peers will be set to `deny`, and ProxyClient peers will be set to `use-local-ip`. When `use-local-ip` is specified, both ProxySG and ProxyClient Branch peers will be set to `use-local-ip`. The local IP is the IP address of the Concentrator ProxySG.

`SGOS#(config adn tunnel) reflect-client-ip peer-sg {allow | deny | use-local-ip}`
Determines the behavior of the ADN Concentrator peer when a ProxySG Branch peer requests client IP reflection for an inbound tunnel connection. The `allow` option allows the request and reflects the client IP. The `deny` option rejects the request and the connection. The `use-local-ip` option allows the connection but uses the IP address of the Concentrator peer.

`SGOS#(config adn tunnel) reflect-client-ip proxy-client {allow | deny | use-local-ip}`
Determines the behavior of the ADN Concentrator peer when a ProxyClient peer requests client IP reflection for an inbound tunnel connection. The `allow` option allows the request and reflects the client IP. The `deny` option rejects the request and the connection. The `use-local-ip` option allows the connection but uses the IP address of the Concentrator peer.

`SGOS#(config adn tunnel) secure-port` *`port_number`*
Configure listening port for secure ADN tunnel

`SGOS#(config adn tunnel) tcp-window-size {auto | size_in_bytes}`
Sets the TCP window size for ADN optimization tunnel connections based on current network conditions and on the receiving host's acknowledgement. `Auto` is the default; under most circumstances, this option should not be set manually.

`SGOS#(config adn tunnel) view`
Views the current configuration ADN tunnel parameters.

`SGOS#(config adn) view`
Views the configuration of the WAN optimization parameters you created on this system.

For More Information

- *SGOS Administration Guide*, Configuring Application Delivery Network chapter

Example

```
SGOS#(config adn)
SGOS#(config adn) enable
SGOS#(config adn) manager
SGOS#(config adn manager) primary-manager 2001:418:9804:111::169
SGOS#(config adn) backup-manager 10.25.36.48
SGOS#(config adn) tunnel
SGOS#(config adn tunnel) tcp-window-size 200000
SGOS#(config adn tunnel) exit
SGOS#(config adn) routing
SGOS#(config adn routing) server-subnets
SGOS#(config adn routing server-subnets) clear-all
SGOS#(config adn routing server-subnets) add 10.9.59.0/24
SGOS#(config adn routing server-subnets) add 2001:418:9804:100::84/128
SGOS#(config adn routing server-subnets) exit
SGOS#(config adn routing) exit
SGOS#(config adn) byte-cache
SGOS#(config adn byte-cache) max-peer-memory 40
SGOS#(config adn byte-cache) exit
```

```
SGOS#(config adn) view
Application Delivery Network Configuration:
ADN:                      enabled
External VIP:             none

Manager Configuration:
Primary manager:          self
Backup manager:           none
Port:                     3034
Secure port:              3036
Approved device           Connecting from
Allow pending devices:    enabled
Pending device            Connecting from
Byte-cache Configuration:
Max number of peers:      10347
Max peer memory:          30

Tunnel Configuration:
Port:                     3035
Secure port:              3037
Bypass if no concentrator:disabled
proxy-processing http:    disabled
accept-transparent:       enabled
connect-transparent:      enabled
last-peer-detection:      enabled
preserve-dest-port:       enabled
TCP window size:          65536
reflect-client-ip peer-sg:          use-local-ip
reflect-client-ip proxy-client:     use-local-ip
Preferred IP Addresses:
<None>
Routing Configuration:
Internet Gateway:         disabled
Exempt Server subnet:     10.0.0.0/8
Exempt Server subnet:     172.16.0.0/12
Exempt Server subnet:      192.168.0.0/16
Exempt Server subnet:     fe80::/10
Exempt Server subnet:     fc00::/7

Security Configuration:
Device-auth-profile:      bluecoat
Manager-listening mode:   plain-only
Tunnel-listening mode:    plain-only
Authorization:            enabled
Secure-outbound:          none
```

#(config) alert

Synopsis

Configures the notification properties of hardware environmental metrics (called *sensors*) and the threshold and notification properties of system resource health monitoring metrics. These *health monitoring* metrics allow you to assess the health of the ProxySG.

Note: Sensor thresholds are not configurable.

Syntax

```
#(config) alert threshold metric_name warning_threshold warning_interval
critical_threshold critical_interval
#(config) alert notification metric_name notification_method
#(config) alert severity sensor power-supply condition
```

Subcommands

Threshold

`#(config) alert threshold cpu-utilization {warn-threshold | warn-interval | crit-threshold | crit-interval}`
Sets alert threshold properties for CPU utilization metrics.

`#(config) alert threshold icap deferred-connections <crit-threshold>}`
Sets alert threshold properties for deferred ICAP connections. The `crit-threshold` value is a percentage between 1 and 1000, (80% default) based on the number of ICAP connections in a deferred state.

`#(config) alert threshold icap queued-connections <crit-threshold>}`
Sets alert threshold properties for ICAP queued connections. The `crit-threshold` value is a percentage between 1 and 1000 based on the number of connections queued awaiting an available ICAP connection.

`#(config) alert threshold license-utilization {warn-threshold | warn-interval | crit-threshold | crit-interval}`
Sets alert threshold properties for licenses with user limits.

`#(config) alert threshold license-expiration {sgos {warn-threshold | warn-interval | crit-threshold | crit-interval} | ssl {warn-threshold | warn-interval | crit-threshold | crit-interval}}`
Sets alert threshold properties for license expiration.

`#(config) alert threshold memory-utilization {warn-threshold | warn-interval | crit-threshold | crit-interval}`
Sets alert threshold properties for memory pressure metrics.

`#(config) alert threshold network-utilization adapter[:interface]{warn-threshold | warn-interval | crit-threshold | crit-interval}`
Sets alert threshold properties for interface utilization metrics.

`#(config) alert threshold cloud-common-policy {entitlement {warn-threshold | crit-threshold} update-errors {warn-threshold | warn-interval | crit-threshold}}`
Sets alert threshold properties for cloud common policy entitlement and update errors. All settings revert to defaults if the appliance is deregistered from the cloud service.

`#(config) alert notification adn {connection | manager}`
Sets alert notification properties for ADN.

`#(config) alert notification cpu-utilization {email | log | trap | none}`
Sets alert notification properties for cpu utilization metrics.

`#(config) alert notification disk-status {email | log | trap | none}`
Sets alert notification properties for disk status messages.

`#(config) alert notification failover {email | log | trap | none}`
Sets alert notification properties for failover partners. If a failover occurs, notification is sent by the new master.

`#(config) alert notification icap deferred-connections {email | log | trap | none}`
Sets alert notification properties for deferred ICAP connections. When the percentage of deferred ICAP connections exceeds the threshold defined with `alert threshold icap deferred-connections`, the ProxySG Appliance will output a message via the configured method. When the number of connections decreases below the configured threshold, another log entry is added to the specified output.

`#(config) alert notification icap queued-connections {email | log | trap | none}`
Sets alert notification output for ICAP connections that exceed the number of available connections configured in the ICAP service, based on the percentage configured with a`lert threshold icap queued-connections`. When the number of queued ICAP connections drops below the configured threshold, another log entry is added to the specified output..

`#(config) alert notification health-check {email | log | trap | none}`
Sets alert notification properties for health-checks globally.

`#(config) alert notification license-utilization users {email | log | trap | none}`
Sets alert notification properties for licenses with user limits.

`#(config) alert notification license-expiration {sgos {email | log | trap | none} | ssl {email | log | trap | none}}`
Sets the alert notification properties for SGOS or SSL license expiration.

`#(config) alert notification memory-utilization {email | log | trap | none}`
Sets the notification alert properties for memory utilization.

`#(config) alert notification network-utilization` *adapter*`[:`*interface*`]{email | log | trap | none}`
Sets the alert notification properties for network utilization.

`#(config) alert notification cloud-common-policy {entitlement {email | log | trap | none} | update-errors {email | log | trap | none}}`
Sets the alert notification properties for cloud common policy and related synchronization update errors. Set the e-mail properties using the `event-log mail` command. All settings revert to defaults if the appliance is deregistered from the cloud service.

`#(config) alert notification reboot {email | log | trap | none}`
Sets the alert notification properties for system reboot. When this command is set and the system reboots, a reboot notification is sent by e-mail, event log, SNMP trap, or a combination of these. Set the e-mail properties using the `event-log mail` command. If `email` is set but not `log`, the reboot will still be logged.

`#(config) alert notification sensor {fan {email | log | trap | none} | power-supply {email | log | trap | none} | temperature {email | log | trap | none} | voltage {email | log | trap | none}}`
Sets alert notification properties for hardware environmentals. See "Sensors" on page 118 for a description of the sensor types.

`#(config) alert severity sensor power-supply {critical | no-effect | warning}`
Sets the severity level for an undetected power-supply.

Sensors

The following table describes the sensor metrics. The hardware and environmental metrics are referred to as sensors. Sensor threshold values are not configurable and are preset to optimal values. For example, if the CPU temperature reaches 55 degrees Celsius, it is considered to have entered the Warning threshold.

Table 3-1. Sensor Health Monitoring Metrics

Metric	MIB	Threshold States

Disk status	Disk	Critical:
		`Bad`
		Warning:
		`Removed`
		`Offline`
		OK:
		`Present`
		`Not Present`

Temperature Bus temperature CPU temperature	Sensor	Critical Warning OK
Fan CPU Fan Voltage Bus Voltage CPU voltage Power Supply voltage	Sensor Sensor	Critical Warning OK Critical Warning OK

Thresholds

The following table describes the health monitoring metrics and default thresholds. Sensor thresholds cannot be configured.

Table 3-2. System Resource Health Monitoring Metrics

Metric	Units	Threshold and Interval Defaults	Notes
CPU Utilization	Percentage	Critical: 95/120 Warning: 80/120	Measures the value of CPU 0 on multi-processor systems--*not* the average of all CPU activity.
Memory Utilization	Percentage	Critical: 95/120 Warning: 90/120	Memory pressure occurs when memory resources become limited, causing new connections to be delayed.
Network Utilization	Percentage	Critical: 90/120 Warning: 60/120	Measures the traffic (in and out) on the interface to determine if it is approaching the maximum allowable bandwidth.
License Utilization	Percentage	Critical: 90/0 Warning: 80/0	For licenses that have user limits, monitors the number of users.

Table 3-2. System Resource Health Monitoring Metrics (Continued)

Metric	Units	Threshold and Interval Defaults	Notes
SGOS Base and SSL Proxy License Expiration	Days	Critical: 0/0 Warning: 15/0 (For new ProxySG appliances running SGOS 5.3)	Warns of impending license expiration. For license expiration metrics, intervals are ignored.
Cloud Services: Common Policy Entitlement	0 days / 0	Critical: 0/0 Warning: 30/0	Warns of impending entitlement expiration. For license expiration metrics, intervals are ignored.

For the purposes of notification, thresholds are defined by two variables, the *threshold level* and the *threshold interval*:

- The threshold level describes the state of the metric: OK, Warning, or Critical.

Note: Sensors have different threshold levels than OK, Warning, and Critical. See "Sensors" on page 118 for more information.

- The threshold interval specifies the period of time that the metric must stay in the level before an alert is triggered.

Consider the following command:

```
#(config) alert threshold cpu-utilization 80 20 90 20
```

The preceding command sets the `cpu-utilization` threshold values as follows:

- Warning Threshold=80 (percent)
- Warning Interval=20 (seconds)
- Critical Threshold=90 (percent)
- Critical Interval=20 (seconds)

In this example, if CPU activity hovers between 80% and 89% for 20 seconds, the cpu-utilization metric is considered to be in the Warning condition.

Notification occurs when a threshold state changes, for example, from OK to Warning. See "Notification Methods" on page 120 for more information.

Notification Methods

The following notification methods can be set. To set more than one type of notification, separate the notification method by spaces. For example:

```
#(config)alert notification license-utilization users email log trap
```

Table 3-3. Alert Notification Methods

Method	Description
`email`	Notify using e-mail (set in the `event-log mail` command)
`log`	Notify using Event log

Table 3-3. Alert Notification Methods (Continued)

Method	Description
`trap`	Notify using SNMP trap
`none`	Disable notification

Licenses

The license utilization and expiration alert settings can be modified for the following licenses.

Table 3-4. Health Monitoring License Options

Method.	Description
`sgos`	Alert properties for SGOS (expiration only)
`ssl`	Alert properties for SSL Proxy (expiration only)
`cloud services common-policy`	Alert properties for cloud common policy (expiration only)

The threshold values for license expiration metrics are set in days until expiration. In this context, a "critical" threshold indicates that license expiration is imminent. This is the only metric in which the Critical threshold value should be smaller than the Warning threshold value. For example, if you set the Warning threshold to 45, an alert is sent when there are 45 days remaining in the license period. The Critical threshold would be less than 45 days, for example 5 days.

For the license expiration metrics, the threshold interval is irrelevant and is set by default to 0. The Critical threshold is configured to 0, which means that a trap is immediately sent upon license expiration.

You should set the Warning Threshold to a value that gives you ample time to renew your license. For new ProxySG appliances running SGOS 5.3, the default Warning threshold for license expiration is 15 days. For ProxySG appliances upgrading from earlier versions to SGOS 5.3, the default Warning threshold remains at the same value prior to the upgrade. For example, if the Warning threshold was 30 days prior to the upgrade, the Warning threshold will remain at 30 days after the upgrade. Refer to the most current Release Notes for SGOS upgrade information.

For More Information

- *SGOS Administration Guide*

Examples

```
#(config) alert threshold cpu-utilization 80 20 90 20
#(config) alert threshold license-utilization users 80 20 90 20
#(config) alert threshold license-expiration sgos 65 30
#(config) alert notification cpu-utilization trap
#(config) alert notification license-utilization users email log trap
#(config) alert notification sensor fan email
#(config) alert notification sensor voltage trap
```

#(config) appliance-name

Synopsis

Use this command to assign a name to a ProxySG appliance. Any descriptive name that helps identify the system is sufficient.

Syntax

`#(config)` **`appliance-name`** *`name`*

Associates *`name`* with the current ProxySG.

For More Information

- *SGOS Administration Guide*

Example

```
SGOS#(config) appliance-name ProxyDev1
 ok
```

#(config) application-protection

Synopsis

Allows you to configure the Application Protection service.

Syntax

```
#(config) application-protection
```

This enters application-protection mode and changes the prompt to:

```
#(config application-protection)
```

Subcommands

`#(config application-protection)` **`disable`**

Disables the Application Protection service. When you issue this command, it checks if SQL injection detection is enabled in policy. If it is enabled, the CLI warns that disabling the Application Protection service also disables the SQL injection detection policy.

`#(config application-protection)` **`download get-now`**

Initiates an immediate download of the application protection database. If errors occur during subscription content processing, the CLI displays a message indicating the reason for the failure.

`#(config application-protection)` **`download notify-only disable`**

Disables the notify-only setting.

`#(config application-protection)` **`download notify-only enable`**

When a new database version is available for download, a notification is sent to the administrator and also recorded in the event log. You can use this setting only after the first successful database download. Use this setting in a test environment.

`#(config application-protection)` **`enable`**

Enables the Application Protection service.

`#(config application-protection)` **`exit`**

Exits application-protection mode and returns to the `#(config)` prompt.

`#(config application-protection)` **`view`**

Displays the current Application Protection service settings, including download status.

Example

Enable the notify-only setting for use in a test environment.

```
#(config application-protection)download notify-only enable
  ok
```

#(config) archive-configuration

Synopsis

Archiving a ProxySG system configuration on a regular basis is always a good idea. In the rare case of a complete system failure, restoring a ProxySG to its previous state is simplified by loading an archived system configuration from an FTP, HTTP, or HTTPS server. The archive contains all system settings differing from system defaults, along with any forwarding and security lists installed on the ProxySG.

Unless you restore the SSL `configuration-passwords-key` keyring from the source device, archives can only be restored onto the same device that was the source of the archive. This is because the encrypted passwords in the configuration (login, enable, FTP, etc.) cannot be decrypted by a device other than that on which it was encrypted.

Syntax

`#(config) archive-configuration [subcommands]`

Subcommands

`#(config) archive-configuration archive-signing {enforce-signed {enable | disable} | signing-keyring {keyring-name} | verify-ccl {ccl-name}}`
Configures the archiving signing options. A signed archive is a configuration backup that is cryptographically signed with a key known only to the signing entity—the digital signature guarantees the integrity of the content and the identity of the originating device. You can then use a trusted CA Certificate List (CCL) to verify the authenticity of the archive.

The enforce-signed option enforces installation of only signed archives. The `signing-keyring` option specifies the keyring that will be used to sign archives. The `verify-ccl` option specifies the CCL to use for verifying signed archives.

`#(config) archive-configuration encrypted-password encrypted_password`
Encrypted password for upload host (not required for TFTP)

`#(config) archive-configuration filename-prefix filename`
Specifies the prefix that should be applied to the archive configuration on upload. For example, %H (Hour in 24-hour format). Refer to the "Backing Up the Configuration" chapter in the *SGOS Administration Guide* for a complete list of file name prefixes.

`#(config) archive-configuration host hostname`
Specifies the HTTP, HTTPS, FTP, or TFTP host to which the archive configuration should be uploaded. The *hostname* can be an IPv4 or IPv6 address, or a domain name that resolves to an IPv4 or IPv6 address. If an IPv6 address is specified for the *hostname*, it must be enclosed in brackets, for example:
`archive-configuration host [2001:db8:85a3::8a2e:370:7334]`

`#(config) archive-configuration no signing-keyring`
Disables the requirement for signed archives.

`#(config) archive-configuration password password`
Specifies the password for the host to which the archive configuration should be uploaded

`#(config) archive-configuration path path`
Specifies the path to the HTTP, HTTPS, or FTP host to which the archive configuration should be uploaded. Not required for TFTP.

`#(config) archive-configuration port port`
Specifies the port to use for uploading the archive.

`#(config) archive-configuration protocol {ftp | tftp | http | https}`
Uploads the archive using the specified protocol—HTTP, HTTPS, FTP, or TFTP.

`#(config)` **`archive-configuration ssl-device-profile`** *`ssl-device-profile name`*
Specifies the device profile used for SSL connections. An SSL device profile contains the information required for device authentication, including the name of the keyring that contains the private key and certificate this device uses to authenticate itself. The default keyring is appliance-key.

`#(config)` **`archive-configuration username`** *`username`*
Specifies the username for the remote host to which the archive configuration should be uploaded. Not required for TFTP.

For More Information

- *SGOS Administration Guide*

Example

```
SGOS#(config) archive-configuration host host3
 ok
```

#(config) asymmetric-route-bypass

Synopsis

When `reflect-client-ip` is enabled, the ProxySG is able to detect asymmetric routing for intercepted connections. New connections from the same source and destination IP pair are dynamically bypassed after asymmetric routing is detected (detection occurs on the first reset packet). The IP pairs are added to a table that contains the list of dynamically bypassed asymmetric routes.

Syntax

```
#(config) asymmetric-route-bypass
```

This changes the prompt to:

```
#(config asymmetric-route-bypass)
```

Subcommands

`#(config asymmetric-route-bypass) clear`
Clears all asymmetric route entries.

`#(config asymmetric-route-bypass) [disable | enable]`
Disables/enables asymmetric route detection.

`#(config asymmetric-route-bypass) exit`
Leaves `#(config asymmetric-route-detection)` mode and returns to `#(config)` mode.

`#(config asymmetric-route-bypass) max-entries` *number_of_entries*
Set maximum number of entries allowed in the asymmetric route bypass list.

`#(config asymmetric-route-bypass) remove [*` | *source_ip*`] [*` | *destination_ip*`]`
Remove an asymmetric bypass route.

`#(config asymmetric-route-bypass) server-threshold` *entries*
Configure threshold to trigger consolidation of entries.

`#(config asymmetric-route-bypass) timeout` *minutes*
Set the expiration timeout.

`#(config asymmetric-route-bypass) view`
View the current configuration.

#(config) attack-detection

Synopsis

The ProxySG can reduce the effects of distributed denial of service (DDoS) attacks and port scanning, two of the most common virus infections.

The ProxySG prevents attacks by limiting the number of TCP connections from each client IP address and either will not respond to connection attempts from a client already at this limit or will reset the connection.

Syntax

```
#(config) attack-detection
```

This changes the prompt to:

```
#(config attack-detection)
```

Subcommands

`#(config attack-detection) client`
Changes the prompt to **#(config client)** on page 129.

`#(config attack-detection) exit`
Leaves `#(config attack-detection)` mode and returns to `#(config)` mode.

`#(config attack-detection) server`
Changes the prompt to **#(config server)** on page 132.

`#(config attack-detection) view client [blocked | connections | statistics]`
Displays client information. The `blocked` option displays the clients blocked at the network level, the `connections` option displays the client connection table, and the `statistics` option displays client request failure statistics.

`#(config attack-detection) view configuration`
Allows you to view attack-detection configuration settings or the number of current connections.

`#(config attack-detection) view server statistics`
Displays server information. The statistics option displays server-connection failure statistics

For More Information

- *SGOS Administration Guide*

Example

```
#(config attack-detection) view configuration
Client limits enabled:            false
Client interval:                  20 minutes
Default client limits:
Client connection limit:          100
Client failure limit:             50
Client warning limit:             10
Blocked client action:            Drop
Client connection unblock time:   unlimited
```

```
Client limits for 10.9.59.210:
Client connection limit:          100
Client failure limit:             50
Client warning limit:             10
Blocked client action:            Drop
Client connection unblock time:   unlimited
```

#(config client)

Synopsis

Configures a client for attack detection.

Syntax

```
#(config attack-detection) client
```

This changes the prompt to

```
#(config client)
```

Subcommands

`#(config client) block` *ip_address* [*minutes*]
Blocks a specific IP address for the number of minutes listed. If the optional minutes argument is omitted, the client is blocked until explicitly unblocked.

`#(config client) concurrent-request-limit` *integer_between_1_and_2147483647*
Indicates the maximum number of simultaneous requests that effective client IP sources (with `client.effective_address` policy) or explicit client IP sources (without `client.effective_address` policy) are allowed to make. The default value is unlimited.

`#(config client) create` *ip_address or ip_address_and_length*
Creates a client with the specified IP address or subnet.

`#(config client) default {block-action {drop | send-tcp-rst} | connection-limit` *number_of_tcp_connections* `| failure-limit` *number_of_requests* `| unblock-time` *minutes* `| warning-limit` *number_of_warnings*`}`
Default indicates the values that are used if a client does not have specific limits set. These settings can over overridden on a per-client basis.
If they are modified on a per-client basis, the specified limits become the default for new clients. To change the limits on a per-client basis, see *edit*, below.

System defaults for attack-detection limits are:

- block-action: drop
- connection-limit: 100
- failure-limit: 50
- unblock-time: unlimited
- warning-limit: 10

`#(config client) delete` *ip_address or ip_address_and_length*
Deletes the specified client.

`#(config client) {disable-limits | enable limits}`
Enables (sets to true) or disables (sets to false) attack detection.

`#(config client) edit` *ip_address*
Changes the prompt to `#(config client` *ip_address*`)`.

`#(config client` *IP_address*`) block-action {drop | send-tcp-rst}`
Indicates the behavior when the client is at the maximum number of connections or exceed the warning limit: drop connections that are over the limit or send TCP RST for connections over the limit. The default is drop.

`#(config client` *IP_address*`) connection-limit` *number_of_tcp_connections*
Indicates the number of simultaneous connections between 1 and 65535. The default is 100.

`#(config client IP_address)` **`exit`**
Exits the **`#(config client`** *`ip_address`***`)`** submode and returns to `#(config client)` mode.

`#(config client IP_address)` **`failure-limit`** *`number_of_requests`*
Indicates the maximum number of failed requests a client is allowed before the proxy starts issuing warnings. Default is 50. This limit can be modified on a per-client basis.

`#(config client IP_address)` **`monitor-only`**
Enables monitor-only mode, which logs the defined thresholds that have been exceeded, but does not enforce the rules. The default value is disabled. This limit can be modified on a per-client basis.

Note: The monitor-only mode setting has a higher precedence level than the default enforce mode. Enabling monitor-only mode disables rule enforcement.

`#(config client IP_address)` **`no`** {**`connection-limit`** | **`failure-limit`** | **`warning-limit`** | **`unblock-time`**}
Clears the specified limits on a per-client basis. If you edit an existing client's limits to a smaller value, the new value only applies to new connections to that client. For example, if the old value was 10 simultaneous connections and the new value is 5, existing connections above 5 are not dropped.

`#(config client IP_address)` **`request-limit`** *`integer_between_1_and_2147483647`*
Indicates the maximum number of HTTP requests that IP sources are allowed to make during a one-minute interval. The default value is unlimited. This limit can be applied on a per-client basis.

`#(config client IP_address)` **`unblock-time`** *`minutes`*
Indicates the amount of time a client is blocked at the network level when the client-warning-limit is exceeded. Time must be a multiple of 10 minutes, up to a maximum of 1440. The default is unlimited.

`#(config client IP_address)` **`view`**
Displays the limits for this client.

`#(config client IP_address)` **`warning-limit`** *`number_of_warnings`*}
Indicates the number of warnings sent to the client before the client is blocked at the network level and the administrator is notified. The default is 10; the maximum is 100.

`#(config client IP_address)` **`enable-limits`**
Enables attack detection. This is a global setting and cannot be configured individually for specific clients.

`#(config client IP_address)` **`interval`** *`minutes`*
Indicates the amount of time, in multiples of 10 minutes, that client activity is monitored. The default is 20. Note that this is a global limit and cannot be modified for individual clients.

`#(config client IP_address)` **`no default`** {**`connection-limit`** | **`failure-limit`** | **`warning-limit`** | **`unblock-time`**}
Clears the specified limit settings. These settings are applied to all new clients.

`#(config client IP_address)` **`view`** [**`blocked`** | **`connections`** | **`statistics`**]
Views all limits for all clients, or you can show clients blocked at the network level, view the client connection table, or view client request failure statistics.

`#(config client IP_address)` **`unblock`** *`ip_address`*
Releases a specific IP address.

`#(config client)` **`exit`**
Exits the **`#(config client)`** submode and returns to `#(config attack-detection)` mode.

`#(config client)` **`interval`** *`minutes`*
Indicates the amount of time, in multiples of 10 minutes, that client activity is monitored. The default is 20. Note that this is a global limit and cannot be modified for individual clients.

`#(config client) no default {connection-limit | failure-limit | warning-limit | unblock-time}`
Clears the specified limit settings. These settings are applied to all new clients.

`#(config client) view [blocked | connections | statistics]`
Views all limits for all clients, or you can show clients blocked at the network level, view the client connection table, or view client request failure statistics.

`#(config client) unblock ip_address`
Releases a specific IP address.

For More Information

- *SGOS Administration Guide*

Example

```
SGOS#(config) attack-detection
SGOS#(config attack-detection) client
SGOS#(config client) view
Client limits enabled:            true
Client interval:                  20 minutes
Default client limits:
Client connection limit:          700
Client failure limit:             50
Client warning limit:             10
Blocked client action:            Drop
Client connection unblock time:   unlimited
Client limits for 10.9.17.159:
Client connection limit:          unlimited
Client failure limit:             unlimited
Client warning limit:             unlimited
Blocked client action:            Drop
Client connection unblock time:   unlimited
Client limits for 10.9.17.134:
Client connection limit:          700
Client failure limit:             50
Client warning limit:             10
Blocked client action:            Drop
Client connection unblock time:   unlimited
```

#(config server)

Synopsis

Configures a server for attack detection.

Syntax

```
#(config attack-detection) server
```

This changes the prompt to:

```
#(config server)
```

Subcommands

`#(config server)` **`create`** *`hostname`*
Creates a server or server group that is identified by the hostname.

`#(config server)` **`delete`** *`hostname`*
Deletes a server or server group.

`#(config server)` **`edit`** *`hostname`*
Modifies the limits for a specific server.

`#(config server)` **`exit`**
Exits the `#(config server)` submode and returns to `#(config attack-detection)` mode.

`#(config server)` **`view`** `[`**`statistics`**`]`
Displays the request limit for all servers or server groups.

For More Information

- *SGOS Administration Guide*

Example

```
SGOS#(config) attack-detection
SGOS#(config attack-detection) server
SGOS#(config server) create test1
ok
SGOS#(config server) edit test1
SGOS#(config server test1) add 10.9.17.134
ok
SGOS#(config server test1) view
Server configuration for test1:
Request limit:  1000
Host:           10.9.17.134
```

#(config) background-dns-updates

Synopsis

Background DNS updates allows configuration of background DNS updates used in forwarding systems.

Syntax

`#(config) background-dns-updates [subcommands]`

Subcommands

`#(config) background-dns-updates failure-interval seconds`
Sets the seconds between DNS resolution attempts when DNS failures.

`#(config) background-dns-updates maximum-ttl {none | seconds}`
Disables or sets the maximum seconds allowed before the next DNS resolution attempt.

`#(config) background-dns-updates minimum-ttl seconds`
Sets the minimum seconds allowed before the next DNS resolution attempt.

For More Information

- *SGOS Administration Guide*

Example

```
SGOS#(config) background-dns-updates failure-interval 100
 ok
```

#(config) bandwidth-gain

Synopsis

Bandwidth gain is a measure of the effective increase of server bandwidth resulting from the client's use of a content accelerator. For example, a bandwidth gain of 100% means that traffic volume from the ProxySG to its clients is twice as great as the traffic volume being delivered to the ProxySG from the origin server(s). Using bandwidth gain mode can provide substantial gains in apparent performance.

Keep in mind that bandwidth gain is a relative measure of the ProxySG's ability to amplify traffic volume between an origin server and the clients served by the device.

Syntax

`#(config) bandwidth-gain disable`
Disables bandwidth-gain mode

`#(config) bandwidth-gain enable`
Enables bandwidth-gain mode.

For More Information

- *SGOS Administration Guide*

Example

```
SGOS#(config) bandwidth-gain enable
 ok
```

#(config) bandwidth-management

Synopsis

Bandwidth management allows you to classify, control, and, if required, limit the amount of bandwidth used by a class of network traffic flowing into or out of the ProxySG.

Syntax

```
#(config) bandwidth-management
```

This changes the prompt to:

```
#(config bandwidth-management)
```

Subcommands

`#(config bandwidth-management)` **`create`** *`class_name`*
Creates a bandwidth-management class.

`#(config bandwidth-management)` **`delete`** *`class_name`*
Deletes the specified bandwidth-management class. Note that if another class has a reference to the specified class, this command fails.

`#(config bandwidth-management)` **`disable`**
Disables bandwidth-management.

`#(config bandwidth-management)` **`edit`** *`class_name`*—changes the prompt (see **`#(config bandwidth-management class_name)`** on page 136)

`#(config bandwidth-management)` **`enable`**
Enables bandwidth-management.

`#(config bandwidth-management)` **`exit`**
Exits `#(config bandwidth-management)` mode and returns to `#(config)` mode.

`#(config bandwidth-management)` **`view configuration`** [*`bandwidth_class`*]
Displays bandwidth-management configuration for all bandwidth-management classes or for the class specified.

`#(config bandwidth-management)` **`view statistics`** [*`bandwidth_class`*]
Displays bandwidth-management statistics for all bandwidth-management classes or for the class specified.

For More Information

- *SGOS Administration Guide*

Example

```
SGOS#(config) bandwidth-management
SGOS#(config bandwidth-management) enable
 ok
SGOS#(config bandwidth-management) create Office_A
 ok
SGOS#(config bandwidth-management) edit Office_A
SGOS#(config bw-class Office_A) exit
SGOS#(config bandwidth-management) exit
SGOS#(config)
```

#(config bandwidth-management *class_name)*

Synopsis

This command allows you to edit a bandwidth-management class.

Syntax

```
#(config) bandwidth-management
```

This changes the prompt to:

```
#(config bandwidth-management)
#(config bandwidth-management) edit class_name
```

This changes the prompt to:

```
#(config bw-class class_name)
```

Subcommands

`#(config bw-class class_name)` **`exit`**
Exits `#(config bw-class class_name)` mode and returns to `#(config bandwidth-management)` mode.

`#(config bw-class class_name)` **`max-bandwidth`** *`maximum_in_kbps`*
Sets the maximum bandwidth for this class.

`#(config bw-class class_name)` **`min-bandwidth`** *`minimum_in_kbps`*
Sets the minimum bandwidth for this class

`#(config bw-class class_name)` **`no max-bandwidth`**
Resets the maximum bandwidth of this bandwidth-management class to the default (unlimited—no maximum)

`#(config bw-class class_name)` **`no min-bandwidth`**
Resets the minimum bandwidth of this bandwidth-management class to the default (no minimum).

`#(config bw-class class_name)` **`no parent`**
Clears the parent from this bandwidth-management class.

`#(config bw-class class_name)` **`parent`** *`class_name`*
Makes the specified class a parent of the class being configured.

`#(config bw-class class_name)` **`priority`** *`value_from_0_to_7`*
Sets the priority for this bandwidth-management class. The lowest priority level is 0 and the highest is 7.

`#(config bw-class class_name)` **`view`** `[`*`children`*`]`
Displays the settings for this bandwidth-management class or displays the settings for the children of this bandwidth-management class.

For More Information

- *SGOS Administration Guide*

Example

```
SGOS#(config) bandwidth-management
SGOS#(config bandwidth-management) edit CEO_A
SGOS#(config bw-class CEO_A)min-bandwidth 500
 ok
SGOS#(config bw-class CEO_A) priority 1
 ok
SGOS#(config bw-class CEO_A) exit
SGOS#(config bandwidth-management) exit
SGOS#(config)
```

#(config) banner

Synopsis

This command enables you to define a login banner for your users.

Syntax

`#(config)` **`banner login`** *`string`*
Sets the login banner to the value of *`string`*.

`#(config)` **`banner no login`**
Sets the login banner to null.

For More Information

- *SGOS Administration Guide*

Example

```
#(config) banner login “Sales and Marketing Intranet Web”
 ok
```

#(config) bridge

Synopsis

Allows you to configure bridging.

Syntax

```
#(config) bridge
```

This changes the prompt to:

```
#(config bridge)
```

Subcommands

`#(config bridge)` **`bandwidth-class`** *`bridgename`*
Sets bridge bandwidth class.

`#(config bridge)` **`create`** *`bridgename`*
Creates a bridge. This bridge name is case insensitive. You cannot name one bridge "ABC" and another bridge "abc".

`#(config bridge)` **`delete`** *`bridgename`*
Deletes the bridge.

`#(config bridge)` **`edit`** *`bridgename`*
Changes the prompt to `#(config bridge` *`bridgename`*`)`

`#(config bridge` *`bridgename`*`)` **`exit`**
Exits the `#(config bridge` *`hostname`*`)` submode and returns to `#(config bridge)` mode.

`#(config bridge)` **`no bandwidth-class`**
Clears the bandwidth-class settings.

`#(config bridge)` **`view`** {**`configuration`** | **`statistics`** | **`fwtable`**} *`bridgename`*
Displays information for the specified bridge or fall all bridges.

Note: To bandwidth-manage a bridge, bandwidth management must be enabled. Bandwidth management is enabled by default if you have a valid bandwidth-management license. You must also create a bandwidth class for bridging (in bandwidth-management mode) before you can select it here. See **`#(config bandwidth-management class_name)`** on page 136 for more information.

For More Information

- *SGOS Administration Guide*

Example

```
SGOS#(config) bridge
SGOS#(config bridge) create test
 ok
SGOS#(config bridge) exit
SGOS#(config)
```

#(config bridge *bridge_name)*

Synopsis

This command allows you to edit a bridge.

Syntax

```
#(config) bridge
```

This changes the prompt to:

```
#(config bridge)
```

```
#(config bridge) edit bridge_name
```

This changes the prompt to:

```
#(config bridge bridge_name)
```

Subcommands

`#(config bridge bridgename) attach-interface adapter#:interface#`
Attaches the interface to the bridge.

`#(config bridge bridgename) clear-fwtable {static}`
Clears bridge forwarding table.

`#(config bridge bridgename) clear-statistics`
Clears the bridge statistics.

`#(config bridge bridgename) exit`
Exits #(config bridge *bridge_name)* mode and returns to `#(config bridge)` mode.

`#(config bridge bridgename) failover {group | mode} {parallel | serial}`
Associates the bridge to a failover group or sets the bridge failover mode.

`#(config bridge bridgename) mode {disable | fail-open | fail-closed}`
Sets the bridge mode on appliances equipped with a programmable adapter card.

The following adapter card modes are available:

- **`disable:`** Disables the bridge and allows the adapter interfaces to be reused as NICs or as part of another bridge.
- **`fail-open:`** If the ProxySG fails, all traffic passes through the bridge so clients can still receive data.
- **`fail-closed:`** If the ProxySG fails, all traffic is blocked and service is interrupted. This mode provides the same functionality as a user-configured software bridge.

`#(config bridge bridgename) mute {enable | disable}`
Specifies whether to mute the bridge interfaces upon detecting a bridge loop. By default, muting is enabled.

`#(config bridge bridgename) no {interface | failover | static-fwtable-entry}`
Clears the settings as follows:

- `interface`: Removes the interface from the bridge.
- `failover`: Negates failover settings.
- `static-fwtable-entry`: Clears the static forwarding table entry.

`#(config bridge bridgename) spanning-tree adapter#:interface# {enable | disable}`
Enables or disables spanning tree participation.

`#(config bridge` *`bridgename`*`)` **`propagate-failure`** {**`enable`** | **`disable`**}
Enables or disables link error propagation.

`#(config bridge` *`bridgename`*`)` **`static-fwtable-entry`** *`adapter#:interface# mac-address`*
Adds a static forwarding table entry.

`#(config bridge` *`bridgename`*`)` **`mute-on-loop`** {**`enable`** | **`disable`**}
Enable/disable interface muting when a bridge loop is detected. Muting is enabled by default.

`#(config bridge` *`bridgename`*`)` **`view`** {**`configuration`** | **`statistics`** | **`fwtable`**}
Displays information for the specified bridge.

For More Information

- *SGOS Administration Guide*

Example

```
SGOS#(config) bridge
SGOS#(config bridge) edit b_1
SGOS#(config bridge b_1) attach interface 0:1
 ok
SGOS#(config bridge b_1) failover mode parallel
 ok
SGOS#(config bridge b_1) exit
SGOS#(config bridge) exit
SGOS#(config)
```

#(config)cachepulse

Synopsis

Allows you to configure the CachePulse service.

Syntax

```
#(config) cachepulse
```

This changes the prompt to:

```
#(config cachepulse)
```

Subcommands

`#(config cachepulse) disable`
Disables the CachePulse service.

`#(config cachepulse) download get-now`
Initiates an immediate database download.

`#(config cachepulse) enable`
Enables the CachePulse service.

`#(config cachepulse) exit`
Exits the `cachepulse` node and returns to `#(config)` prompt.

`#(config cachepulse) view`
Displays CachePulse statistics, such as license information, registration status, the download URL for the CachePulse database, results of the last download, and the last successful download. This subcommand produces the same output as the **`#show cachepulse`** command.

For More Information

- *SGOS Administration Guide*

Example

```
#(config cachepulse) view
License Type:           Perpetual
Service:                Enabled
Download method:        Direct
Last successful download:
  Time:                 Tue, 30 Jul 2013 20:15:39 UTC
  Downloading from:     https://subscription.es.bluecoat.com/cachepulse/latestPolicy
```

#(config) caching

Synopsis

Objects can be stored and managed for later retrieval.

Discussion

When a stored HTTP object expires, it is placed in a refresh list. The ProxySG processes the refresh list in the background, when it is not serving requests. Refresh policies define how the device handles the refresh process.

The HTTP caching options allow you to specify:

- Maximum object size
- Negative responses
- Refresh parameters

In addition to HTTP objects, the ProxySG can store objects requested using FTP. When the device retrieves and stores an FTP object, it uses two methods to determine how long the object should stay cached.

- If the object has a last-modified date, the ProxySG assigns a refresh date to the object that is a percentage of the last-modified date.
- If the object does not have a last-modified date, the ProxySG assigns a refresh date to the object based on a fixed period of time.

Syntax

```
#(config) caching
```

This changes the prompt to:

```
#(config caching)
```

Subcommands

`#(config caching) always-verify-source`
Specifies the ProxySG to always verify the freshness of an object with the object source.

`#(config caching) exit`
Exits the `#(config caching)` mode and returns to `#(config)` mode.

`#(config caching) ftp`
Changes the prompt to **#(config caching ftp)** on page 145

`#(config caching) max-cache-size` *megabytes*
Specifies the maximum size of the cache to the value indicated by *megabytes*.

`#(config caching) negative-response` *minutes*
Specifies that negative responses should be cached for the time period identified by *minutes*

`#(config caching) no always-verify-source`
Specifies that the ProxySG should never verify the freshness of an object with the object source

`#(config caching) no automatic-backoff`
Disables the HTTP Disk Backoff feature. Enabled by default, this feature monitors disk activity and prevents cache reading or writing during periods of peak activity. If disabled, this feature can be enabled by entering `automatic-backoff` at the `(config caching)` prompt.

`#(config caching) no refresh`
Disables asynchronous adaptive refresh (AAR).

`#(config caching) refresh bandwidth {automatic | kbps}`
Specifies the amount of bandwidth (in kilobits per second) that the ProxySG appliance should use for asynchronous adaptive refresh activity. The range is **0-2097151** kbps; a value of **0** disables adaptive refresh. To have the ProxySG automatically adjust the amount of bandwidth necessary to refresh content, use **automatic**. Asynchronous adaptive refresh is disabled by default.

`#(config caching) view`
Displays caching parameters.

For More Information

- *SGOS Administration Guide*

Example

```
SGOS#(config) caching
SGOS#(config caching) always-verify-source
 ok
SGOS#(config caching) max-cache-size 100
 ok
SGOS#(config caching) negative-response 15
 ok
SGOS#(config caching) refresh bandwidth automatic
 ok
SGOS#(config caching) exit
SGOS#(config)
```

#(config caching ftp)

Synopsis

The FTP caching options allow you to specify:

- Transparency
- Caching objects by date
- Caching objects without a last-modified date: if an FTP object is served without a last modified date, the ProxySG caches the object for a set period of time.

Syntax

```
#(config) caching
```

This changes the prompt to:

```
#(config caching)
```

```
#(config caching) ftp
```

This changes the prompt to:

```
#(config caching ftp)
```

Subcommands

`#(config caching ftp) {disable | enable}`
Disables or enables caching FTP objects

`#(config caching ftp) exit`
Exits `#(config caching ftp)` mode and returns to `#(config caching)` mode.

`#(config caching ftp) type-m-percent` *`percent`*
Specifies the TTL for objects with a last-modified time.

`#(config caching ftp) type-n-initial` *`hours`*
Specifies the TTL for objects with no expiration.

`#(config caching ftp) view`
Shows the current FTP caching settings.

For More Information

- *SGOS Administration Guide*

Example

```
SGOS#(config caching) ftp
SGOS#(config caching ftp) enable
 ok
SGOS#(config caching ftp) type-m-percent 20
 ok
SGOS#(config caching ftp) type-n-initial 10
 ok
SGOS#(config caching ftp) exit
SGOS#(config caching) exit
```

#(config) cifs

Synopsis

Configures the CIFS proxy for SMBv1 connections. See **#(config) smbv2** on page 382 for configuring settings for SMBv2 connections.

Syntax

```
SGOS#(config) cifs
```

This changes the prompt to:

```
SGOS#(config cifs)
```

Subcommands

`SGOS#(config cifs) directory-cache-time` *seconds*
This option determines how long SMBv1 directory information is kept in cache. Changes made to a directory by clients not using the ProxySG may not be visible to ProxySG clients until at least this much time has elapsed. The default cache time is 60 seconds.

`SGOS#(config cifs) disable`
Disable protocol-based acceleration for SMBv1 connections. All SMBv1 connections are passed through, allowing the CIFS proxy to accelerate them with byte caching and compression techniques (if enabled for the CIFS service). No object caching is performed on SMBv1 connections.

`SGOS#(config cifs) enable`
Enable protocol-based acceleration for SMBv1 connections.

`SGOS#(config cifs) exit`
Returns to the (config) submode.

`SGOS#(config cifs) read-ahead {disable | enable}`
This option is enabled by default and improves performance of SMBv1 connections by attempting to fetch and cache blocks of data that might be requested by a client before the actual request occurs. Disabling this option causes the ProxySG to fetch and cache only data actually requested by clients.

`SGOS#(config cifs) remote-storage-optimization {disable | enable}`
When this option is enabled, Windows Explorer modifies the icons of uncached folders on remote servers, indicating to users that the contents of the folder have not yet been cached by the ProxySG. Applies to SMBv1 connections only.

`SGOS#(config cifs) smb-signing domain` *domain*
Configure the domain name to which the username belongs; the ProxySG will use this domain to perform SMB signing. Specifying the domain is optional. SMB signing is supported on SMBv1 connections only.

`SGOS#(config cifs) smb-signing encrypted-password` *encrypted-password*
Specify the encrypted password that the ProxySG sends to access the domain when performing SMB signing. Specifying the encrypted password is optional. SMB signing is supported on SMBv1 connections only.

`SGOS#(config cifs) smb-signing optimize {disable | enable}`
Enable/disable CIFS optimizations on signed SMBv1 traffic. Note: Before enabling SMB signing on the ProxySG, you must create a user in the domain that represents the ProxySG. When SMB signing is required by the OCS, the CIFS proxy uses this virtual user's credentials. This user cannot be a guest or anonymous. SMB signing is supported on SMBv1 connections only.

Note: If the client is configured to *require* SMB signing, which is not a common configuration, the ProxySG cannot provide CIFS optimization; the traffic passes through with only the benefits provided by the general ADN configuration.

`SGOS#(config cifs)` **`smb-signing password`** *`password`*
Specify the user password that the ProxySG sends to access the domain when performing SMB signing. SMB signing is supported on SMBv1 connections only.

`SGOS#(config cifs)` **`smb-signing username`** *`username`*
Specify the user in the domain that will be used to perform SMB signing. Ensure you enter the name exactly as created. Specifying the user name is required. SMB signing is supported on SMBv1 connections only.

`SGOS#(config cifs)` **`strict-directory-expiration {disable | enable}`**
This option is disabled by default. When this option is enabled and `directory-cache-time` is past its expiration, directories are refreshed synchronously instead of in the background. This is needed when the set of visible objects in a directory returned by a server can vary between users.

`SGOS#(config cifs)` **`suppress-folder-customization {disable | enable}`**
To speed the display of remote folders, enable Suppress Folder Customization to skip extra transactions and always display remote folders in the default view.

`SGOS#(config cifs)` **`view {configuration | statistics}`**
Views the configuration or statistics for SMBv1.

`SGOS#(config cifs)` **`write-back {full | none}`**
This option is set to `full` by default, which improves performance by acknowledging client writes immediately and sending them to the server in the background. Setting this option to `none` forces all writes to be sent to the server synchronously.

For More Information

- "Accelerating File Sharing" chapter in the *SGOS Administration Guide*
- **`#(config) smbv2`** on page 382

Example

```
SGOS#(config)cifs
SGOS#(config cifs) directory-cache-time 240
 ok
SGOS#(config cifs) read-ahead enable
 ok
SGOS#(config cifs)write-back full
 ok
SGOS#(config cifs) exit
SGOS#(config)
```

#(config) clock

Synopsis

To manage objects in the cache, a ProxySG must know the current Universal Time Coordinates (UTC) time. By default, the device attempts to connect to a Network Time Protocol (NTP) server to acquire the UTC time. The ProxySG includes a list of NTP servers available on the Internet, and attempts to connect to them in the order they appear in the NTP server list on the NTP tab. If the ProxySG cannot access any of the listed NTP servers, you must manually set the UTC time using the `clock` command.

Syntax

```
#(config) clock [subcommands]
```

Subcommands

`#(config) clock day` *day*
Sets the Universal Time Code (UTC) day to the day indicated by *day*. The value can be any integer from 1 through 31.

`#(config) clock hour` *hour*
Sets the UTC hour to the hour indicated by *hour*. The value can be any integer from 0 through 23.

`#(config) clock minute` *minute*
Sets the UTC minute to the minute indicated by *minute*. The value can be any integer from 0 through 59.

`#(config) clock month` *month*
Sets the UTC month to the month indicated by *month*. The value can be any integer from 1 through 12.

`#(config) clock second` *second*
Sets the UTC second to the second indicated by *second*. The value can be any integer from 0 through 59.

`#(config) clock year` *year*
Sets the UTC year to the year indicated by *year*. The value must take the form *xxxx*.

For More Information

- *SGOS Administration Guide*

Example

```
SGOS#(config) clock year 2003
 ok
SGOS#(config) clock month 4
 ok
SGOS#(config) clock day 1
 ok
SGOS#(config) clock hour 0
 ok
SGOS#(config) clock minute 30
 ok
SGOS#(config) clock second 59
 ok
```

#(config) cloud-service

Synopsis

Use the cloud-service commands to configure options relating to the Blue Coat Cloud Service and Advanced Lab Network (ALN). The Blue Coat Cloud Service enables all subscribed devices to share the same common policy, whether on-premise and off-premise. The policy can also be modified on the appliance to conform to local conditions. In this way, you can create general policies that apply to all locations while overriding rules that conflict with local requirements. To use this service, you must first obtain a Blue Coat Cloud Service account (contact your Blue Coat sales representative).

The ALN provides a Blue Coat cloud service testing environment. It includes all current functionality plus yet-to-be released new features. Use the ALN to preview and test these new features and provide feedback to Blue Coat.

Syntax

```
#(config) cloud-service
```

This changes the prompt to:

```
#(config cloud-service)
```

To view the ALN CLI options, you must use the `reveal-advanced all` command at the `enable` or `config` prompt:

```
#(config)reveal-advanced all
```

For more information, see **# reveal-advanced** on page 77.

Subcommands

#(config cloud-service) **common-policy** {**disable** | **enable**}
Enables or disables subscription to the policy installed on the Blue Coat cloud service. To use this service, Blue Coat WebFilter (BCWF) must be enabled and the appliance must be registered with the cloud service. Enabling the cloud common-policy enables all subscribed devices to share the same policy configuration, whether on-premise and off-premise. The policy synchronizes with the master file every 15 minutes from last boot time. This interval cannot be changed but you can force an immediate update.

#(config cloud-service) **deregister [force]**
Removes the appliance from the Blue Coat cloud service. The `force` option forces deregistration even if there are errors (the appliance removes all cloud-provisioned policy and returns the system to the pre-registration state).

#(config cloud-service) **exit**
Returns to the (config) submode.

#(config cloud-service) **register** *location-name cloud-service-username* [*password*]
Registers the appliance with the Blue Coat cloud service. Before registering the appliance, you must have obtained a Blue Coat cloud service account.

#(config cloud-service) **update-now [force]**
Synchronizes the installed common policy with the master file in the cloud. You can use this command to re-download the common policy even if the ProxySG appliance has the latest copy of policy as this may be useful when troubleshooting.

#(config cloud-service) **cloud-network {advanced-labs | production}**
Selects the cloud service network to use, the Advanced Labs Network (ALN) or production. By default the appliance will always use the production portal. To view this option, you must enter the `reveal-advanced all` command from the config or enable prompt.

Note: You must have an ALN account to use the Advanced Labs Network. To obtain an account, contact your Blue Coat sales representative.

#(config cloud-service) **view**
View Blue Coat cloud service status for the appliance.

For More Information

- *SGOS Administration Guide*

Example

```
SGOS#(config) register chicago2 admin@example.com Admin
ok
SGOS#(config) common-policy enable
ok
SGOS#(config) view
Location:                          chicago2
Last successful update time:       2012-07-11 14:42:04-07:00PDT
Last attempted update time:        2012-07-12 08:03:38-07:00PDT
Failed update attempts:            0
Entitlements:
  Common Policy:                   enabled, expires on 2014-02-28
SGOS#(config) update-now
```

#(config) content

Synopsis

Use this command to manage and manipulate content distribution requests and re-validate requests for HTTP, FTP, CIFS, and streaming content.

Note: The `content` command options are not compatible with transparent FTP.

Syntax

```
#(config) content [subcommands]
```

Subcommands

`#(config) content cancel outstanding-requests`
Specifies to cancel all outstanding content distribution requests and re-validate requests.

`#(config) content cancel url` *`url`*
Specifies to cancel outstanding content distribution requests and re-validate requests for the URL identified by *`url`*.

`#(config) content delete regex` *`regex`*
Specifies to delete content based on the regular expression identified by *`regex`*.

`#(config) content delete url` *`url`*
Specifies to delete content for the URL identified by *`url`*.

`#(config) content distribute` *`url`* `[from` *`from_url`*`]`
Specifies that the content associated with *`url`* should be distributed from the origin server and placed in the ProxySG cache. Specify the `[from` *`from_url`*`]` when users will be accessing content from a different location than what is specified when pre-populating the cache; for example, the `[from` *`from_url`*`]` is useful when you are pre-populating content in a lab environment using a different host from the one that will be used once the appliance is deployed.

To pre-populate a CIFS file, the *`url`* should conform to the following format:

`cifs://`*`domain`*`;`*`username`*`:`*`password`*`@server/share/path-to-file`

To pre-populate HTTP content, the *`url`* should use the following format:

`http://`*`username`*`:`*`password`*`@host:port/path-to-file`

To pre-populate FTP content, the *`url`* should use the following format:

`ftp://`*`username`*`:`*`password`*`@host:port/path-to-file`

To pre-populate streaming content, the *`url`* should use one of the following formats:

`rtsp://`*`username`*`:`*`password`*`@host:port/path-to-file`

`mms://`*`username`*`:`*`password`*`@host:port/path-to-file`

The sub-fields in the URL are subject to the following requirements:

`domain`	• For CIFS content only • Can contain the following characters only: a-z A-Z 0-9 ~!$%&*()-_=+.;',
	Note: Credentials (*`domain;username:password`*) must be supplied in the URL that is being sent to the server. The credentials will be part of the *`url`* field unless the *`from_url`* is specified; in this case, the credentials are specified as part of the *`from_url`*.

username	Can contain the following characters only: a-z A-Z 0-9 ~!$%&*()-_=+'.,
password	Can contain any character except spaces and the following symbols:
server/host	• Can contain the following characters only: a-z A-Z 0-9 ~!$%&*()-_=+;.', • Spaces are not allowed.
share	• For CIFS content only • Can contain any characters except the following: < > : " / \ | ? *
path-to-file	• Can reference a specific file or a directory. If you specify a directory (without a filename), all files and subdirectories in that directory will be pre-populated. • Can contain any characters except the following: < > : " / \ | ? * • If the *path-to-file* contains spaces, enclose the entire URL in quotation marks, or substitute each space with the following escape code: %20. If the *path-to-file* contains a percent sign, substitute the % with %25. **Note for CIFS URLs:** If you do not specify a path, all files and directories in the specified share will be pre-populated.

`#(config) content priority regex` *priority_0-7 regex*
Specifies to add a content deletion policy based on the regular expression identified by *regex*.

`#(config) content priority url` *priority_0-7 url*
Specifies to add a content deletion policy for the URL identified by *url*.

`#(config) content revalidate regex` *regex*
Revalidates the content associated with the regular expression identified by *regex* with the origin server.

`#(config) content revalidate url` *url* `[from` *from_url*`]`
Revalidates the content associated with the *url*.

For More Information

- *Blue Coat Director Configuration and Management Guide*

Example

```
SGOS#(config) content distribute http://www.bluecoat.com
Current time: Mon, 01 Apr 2003 00:34:07 GMT
SGOS#(config) content revalidate url http://www.bluecoat.com
Last load time: Mon, 01 Apr 2003 00:34:07 GMT
SGOS#(config) content distribute http://www.bluecoat.com
Current time: Mon, 01 Apr 2003 00:35:01 GMT
SGOS#(config) content priority url 7 http://www.bluecoat.com
SGOS#(config) content cancel outstanding-requests
SGOS#(config) content delete url http://www.bluecoat.com
```

#(config) content-filter

Synopsis

The ProxySG offers the option of using content filtering to control the type of retrieved content and to filter requests made by clients. The ProxySG supports the following content filtering methods:

- Local database

  This method allows you to create and maintain your own content-filtering list locally, through the ProxySG CLI or Management Console.

- Blue Coat Web Filter (BCWF)

  BCWF is a highly effective content-filtering service that can quickly learn and adapt to the working set of its users. Also, BCWF can use Dynamic Real Time Rating (DRTR) to analyze requested Web pages in real time, blocking new, unrated content on the fly, while providing the database with instant updates that impact all users without service interruption.

- Internet Watch Foundation® (IWF)

  The IWF is a non-profit organization that provides enterprises with a list of known child pornography URLs. The IWF database features a single category called IWF-Restricted, which is detectable and blockable using policy. IWF can be enabled along with other content-filtering services.

- Vendor-based content filtering

  This method allows you to block URLs using vendor-defined categories. For this method, use content-filtering solutions from the following vendors:

  - i-FILTER
  - InterSafe™
  - Optenet
  - Proventia™
  - SurfControl™
  - WebWasher®

  You can also combine this type of content filtering with the ProxySG policies, which use CPL.

- YouTube™

  You can add Blue Coat categories for YouTube and then add policy that refers to these categories to control traffic. For example, you could block videos that YouTube categorizes as Entertainment and Movies. You can enable and disable this feature in the CLI.

  Note: This feature is provided on an "as-is" basis. Blue Coat has no control of, and is not responsible for, information and content provided (or not) by YouTube. You are obligated to comply with all terms of use regarding the foregoing, including quotas that may be imposed by YouTube. Blue Coat shall not be liable for any discontinuance, availability or functionality of the features described herein.

- Denying access to URLs through policy

  This method allows you to block by URL, including filtering by scheme, domain, or individual host or IP address. For this method, you define ProxySG policies, which use CPL.

Syntax

```
#(config) content-filter
```

This changes the prompt to:

```
#(config content-filter)
```

Subcommands

`#(config content-filter)` **`bluecoat`**
Enters configuration mode for Blue Coat Web Filter. See **`#(config bluecoat)`** on page 158.

`#(config content-filter)` **`categories`**
Shows available categories.

`#(config content-filter)` **`cpu-throttle enable | disable`**
Specifies whether to limit CPU utilization to 25% for content filtering database updates. By default, this option is enabled (meaning that CPU is limited for database downloads).

`#(config content-filter)` **`exit`**
Exits configure content filter mode and returns to configure mode.

`#(config content-filter)` **`i-filter`**
Enters configuration mode for i-FILTER. See **`#(config i-filter)`** on page 160.

`#(config content-filter)` **`intersafe`**
Enters configuration mode for InterSafe. See **`#(config intersafe)`** on page 162.

`#(config content-filter)` **`iwf`**
Enters configuration mode for IWF. See **`#(config iwf)`** on page 164.

`#(config content-filter)` **`local`**—changes the prompt (see **`#(config local)`** on page 166)
Enters configuration mode for Local database.

`#(config content-filter)` **`memory-allocation {high | low | normal}`**
Sets the amount of RAM that the content filter service can use.

Note: The default memory allocation (normal) is ideal for most deployments. Changing the memory allocation might have significant impacts on performance of the appliance. Be sure that the setting you choose is appropriate for your deployment.

Content filtering databases are becoming larger and can cause CPU spikes, restarts and issues with the ProxySG appliance's performance. If you find this is the case, you can change the amount of RAM (the ceiling) that the content filtering service (CFS) is allowed to use. The high option maximizes memory use for content-filtering, and the low option minimizes memory use for content- filtering.

Adjust the amount of memory allocated to the database in the following situations:

- If you are not using ADN and have a high transaction rate for content filtering, you can increase the memory allocation setting to `high`. This helps content filtering run more efficiently.
- If you are using both ADN and content filtering but the transaction rate for content filtering is not very high, you can reduce the memory allocation setting to `low`. This makes more resources available for ADN, allowing it to support a larger number of concurrent connections.

The command causes a reload of all enabled content filter providers, with the new effective ceiling in place.

If you downgrade the SGOS, memory allocation reverts to normal for the platform. When re-upgraded, the selected setting is reinstated on the ProxySG appliance.

`#(config content-filter)` **`no review-message`**
Specifies that vendor categorization review be turned off.

`#(config content-filter)` **`optenet`**
Enters configuration mode for Optenet. See **`#(config optenet)`** on page 168.

`#(config content-filter)` **`proventia`**
Enters configuration mode for Proventia. See **`#(config proventia)`** on page 170.

`#(config content-filter)` **`provider bluecoat`** {**`disable`** | **`enable`** | **`lookup-mode`** {**`always`** | **`uncategorized`**}}
Enables or disables `Blue Coat` Web Filter database. The **`lookup-mode`** option specifies whether every URL should be categorized by the downloaded filter.

`#(config content-filter)` **`provider local`** {**`disable`** | **`enable`** | **`lookup-mode`** {**`always`** | **`uncategorized`**}}
Enables or disables a local user database. The **`lookup-mode`** option specifies whether every URL should be categorized by the downloaded filter.

`#(config content-filter)` **`provider iwf`** {**`disable`** | **`enable`** | **`lookup-mode`** {**`always`** | **`uncategorized`**}}
Enables or disables IWF filtering. The **`lookup-mode`** option specifies whether every URL should be categorized by the downloaded filter.

`#(config content-filter)` **`provider 3rd-party i-filter`**
Selects i-FILTER content filtering.

`#(config content-filter)` **`provider 3rd-party intersafe`**
Selects InterSafe content filtering.

`#(config content-filter)` **`provider 3rd-party lookup-mode`**
Sets lookup mode for a 3rd party provider for content filtering.

`#(config content-filter)` **`provider 3rd-party none`**
Specifies that a third-party vendor not be used for content filtering.

`#(config content-filter)` **`provider 3rd-party optenet`**
Selects Optenet content filtering.

`#(config content-filter)` **`provider 3rd-party proventia`**
Selects Proventia Web Filter content filtering.

`#(config content-filter)` **`provider 3rd-party surfcontrol`**
Selects SurfControl content filtering.

`#(config content-filter)` **`provider 3rd-party webwasher`**
Selects Webwasher URL Filter content filtering.

`#(config content-filter)` **`provider youtube`** {**`disable`** | **`enable`**}
Disables or enables Blue Coat categories for YouTube. It is disabled by default.

`#(config content-filter)` **`review-message`**
Used for categorization review for certain Content Filtering vendors.The review-message setting enables two substitutions that can be used in exceptions pages to allow users to review or dispute content categorization results.

`#(config content-filter)` **`surfcontrol`**
Enters configuration mode for SurfControl. See **`#(config surfcontrol)`** on page 172.

`#(config content-filter)` **`test-url`** *`url`*
Displays categories for a URL assigned by the current configuration.

`#(config content-filter)` **`webwasher`**
Enters configuration mode for WebWasher. See **`#(config webwasher)`** on page 174

`#(config content-filter)` **`view`**
Shows the current settings for the local database (if it is in use) and the selected provider (if one is selected).

For More Information

- *SGOS Administration Guide*

Example

```
SGOS#(config) content-filter
SGOS#(config content-filter) provider 3rd-party proventia
loading database....
 ok
SGOS#(config content-filter) exit
SGOS#(config)
```

#(config bluecoat)

Synopsis

Use this command to configure Blue Coat Web Filter content filtering.

Syntax

```
#(config) content-filter
```

This changes the prompt to:

```
#(config content-filter) bluecoat
```

This changes the prompt to:

```
#(config bluecoat)
```

Subcommands

`#(config bluecoat)` **`download all-day`**
Checks for database updates all day.

`#(config bluecoat)` **`download auto`**
Enables automatic database downloads.

`#(config bluecoat)` **`download between-hours`** *`start stop`*
Sets the interval for automatic database update checks.

`#(config bluecoat)` **`download encrypted-password`** *`encrypted_password`*
Specifies the encrypted password for the database download server.

`#(config bluecoat)` **`download get-now`**
Initiates an immediate database download.

`#(config bluecoat)` **`download full-get-now`**
Initiates an immediate database download of the complete BCWF database.

`#(config bluecoat)` **`download password`** *`password`*
Specifies the password for the database download server.

`#(config bluecoat)` **`download url`** {**`default`** | *`url`*}
Specifies using either the default URL or a specific URL for the database download server.

`#(config bluecoat)` **`download username`** *`username`*
Specifies the username for the database download server.

`#(config bluecoat)` **`exit`**
Exits configure bluecoat mode and returns to configure content-filter mode.

`#(config bluecoat)` **`no download auto`**
Disables automatic download.

`#(config bluecoat)` **`no download encrypted-password`**
Clears the encrypted password for the database download server.

`#(config bluecoat)` **`no download password`**
Clears the password for the database download server.

`#(config bluecoat)` **`no download url`**
Clears the URL for the database download server.

`#(config bluecoat)` **`no download username`**
Clears the username for the database download server.

`#(config bluecoat) service {disable | enable}`
Disables or enables the dynamic categorization service.

`#(config bluecoat) service forward {none | host or group_alias}`
Configures the forwarding host for use with dynamic categorization; stops forwarding of dynamic service requests <host-or-group-alias>.

`#(config bluecoat) service secure {enable | disable}`
Configures the securiy of the connection.

`#(config bluecoat) service send-https-url {full | path | disable}`
Configures the HTTPS mode and level of information sent in dynamic categorization requests for HTTPS transactions sent to WebPulse:

- full — Send entire URL (domain, path, and query string).
- path — Send only the domain and path.
- disable — Do not send a rating request for HTTPS transactions.

`#(config bluecoat) service socks-gateway {none | gateway_alias}`
Configures the SOCKS gateway for use with dynamic categorization; stops the use of a SOCKS gateway with dynamic service requests <gateway-alias>.

`#(config bluecoat) service mode {background | realtime | none}`
Configures the default dynamic categorization to run in the background, run in real time, or to not run.

`#(config bluecoat) service send-request-info {enable | disable}`
Configures default dynamic rating service information handling.

`#(config bluecoat) service send-malware-info {enable | disable}`
Configures malware found notifications to the WebPulse service.

`#(config bluecoat) view`
Shows the current Blue Coat content filtering settings.

`#(config bluecoat) view applications`
View supported application names.

`#(config bluecoat) view operations all|<application name>`
View supported application operations.

For More Information

- *SGOS Administration Guide*

Example

```
SGOS#(config) content-filter
SGOS#(config content-filter) bluecoat
SGOS#(config bluecoat) service mode background
 ok
SGOS#(config bluecoat) exit
SGOS#(config content-filter) exit
SGOS#(config)
```

#(config i-filter)

Synopsis

Use this command to configure i-FILTER content filtering

Syntax

```
#(config) content-filter
```

This changes the prompt to:

```
#(config content-filter) i-filter
```

This changes the prompt to:

```
#(config i-filter)
```

Subcommands

`#(config i-filter)` **`download all-day`**
Checks for database updates all day.

`#(config i-filter)` **`download auto`**
Enables automatic database downloads.

`#(config i-filter)` **`download between-hours`** *`start stop`*
Sets the interval for automatic database update checks.

`#(config i-filter)` **`download encrypted-password`** *`encrypted_password`*
Specifies the encrypted password for the database download server.

`#(config i-filter)` **`download get-now`**
Initiates an immediate database download.

`#(config i-filter)` **`download password`** *`password`*
Specifies the password for the database download server.

`#(config i-filter)` **`download url`** {**`default`** | *`url`*}
Specifies using either the default URL or a specific URL for the database download server.

`#(config i-filter)` **`download username`** *`username`*
Specifies the username for the database download server.

`#(config i-filter)` **`exit`**
Exits configure i-filter mode and returns to configure content-filter mode.

`#(config i-filter)` **`no download auto`**
Disables automatic download.

`#(config i-filter)` **`no download encrypted-password`**
Clears the encrypted password for the database download server.

`#(config i-filter)` **`no download password`**
Clears the password for the database download server.

`#(config i-filter)` **`no download url`**
Clears the URL for the database download server.

`#(config i-filter)` **`no download username`**
Clears the username for the database download server.

`#(config i-filter)` **`view`**
Shows the current InterSafe settings.

For More Information

- *SGOS Administration Guide*

Example

```
SGOS#(config) content-filter
SGOS#(config content-filter) i-filter
SGOS#(config i-filter) no download day-of-week mon
 ok
SGOS#(config i-filter) no download day-of-week wed
 ok
SGOS#(config i-filter) exit
SGOS#(config content-filter) exit
SGOS#(config)
```

#(config intersafe)

Synopsis

Use this command to configure InterSafe content filtering.

Syntax

```
#(config) content-filter
```

This changes the prompt to:

```
#(config content-filter) intersafe
```

This changes the prompt to:

```
#(config intersafe)
```

Subcommands

`#(config intersafe)` **`download all-day`**
Checks for database updates all day.

`#(config intersafe)` **`download auto`**
Enables automatic database downloads.

`#(config intersafe)` **`download between-hours`** *`start stop`*
Sets the interval for automatic database update checks.

`#(config intersafe)` **`download encrypted-password`** *`encrypted_password`*
Specifies the encrypted password for the database download server.

`#(config intersafe)` **`download get-now`**
Initiates an immediate database download.

`#(config intersafe)` **`download password`** *`password`*
Specifies the password for the database download server.

`#(config intersafe)` **`download url`** {**`default`** | *`url`*}
Specifies using either the default URL or a specific URL for the database download server.

`#(config intersafe)` **`download username`** *`username`*
Specifies the username for the database download server.

`#(config intersafe)` **`exit`**
Exits configure Intersafe mode and returns to configure content-filter mode.

`#(config intersafe)` **`no download auto`**
Disables automatic download.

`#(config intersafe)` **`no download encrypted-password`**
Clears the encrypted password for the database download server.

`#(config intersafe)` **`no download password`**
Clears the password for the database download server.

`#(config intersafe)` **`no download url`**
Clears the URL for the database download server.

`#(config intersafe)` **`no download username`**
Clears the username for the database download server.

`#(config intersafe)` **`view`**
Shows the current InterSafe settings.

For More Information

- *SGOS Administration Guide*

Example

```
SGOS#(config) content-filter
SGOS#(config content-filter) intersafe
SGOS#(config intersafe) no download day-of-week mon
 ok
SGOS#(config intersafe) no download day-of-week wed
 ok
SGOS#(config intersafe) exit
SGOS#(config content-filter) exit
SGOS#(config)
```

#(config iwf)

Synopsis

Use this command to configure Internet Watch Foundation content filtering.

Syntax

```
#(config) content-filter
```

This changes the prompt to:

```
#(config content-filter) iwf
```

This changes the prompt to:

```
#(config iwf)
```

Subcommands

`#(config iwf)` **`download all-day`**
Checks for database updates all day.

`#(config iwf)` **`download auto`**
Enables automatic database downloads.

`#(config iwf)` **`download between-hours`** *`start stop`*
Sets the interval for automatic database update checks.

`#(config iwf)` **`download encrypted-password`** *`encrypted_password`*
Specifies the encrypted password for the database download server.

`#(config iwf)` **`download get-now`**
Initiates an immediate database download.

`#(config iwf)` **`download password`** *`password`*
(Optional) Specifies the password for the database download server.

`#(config iwf)` **`download url`** {**`default`** | *`url`*}
Specifies using either the default URL or a specific URL for the database download server.

`#(config iwf)` **`download username`** *`username`*
Specifies the username for the database download server.

`#(config iwf)` **`exit`**
Exits configure Intersafe mode and returns to `#(configure content-filter)` mode.

`#(config iwf)` **`no download auto`**
Disables automatic download.

`#(config iwf)` **`no download encrypted-password`**
Clears the encrypted password for the database download server.

`#(config iwf)` **`no download password`**
Clears the password for the database download server.

`#(config iwf)` **`no download url`**
Clears the URL for the database download server.

`#(config iwf)` **`no download username`**
Clears the username for the database download server.

`#(config iwf)` **`view`**
Shows the current InterSafe settings.

Example

```
SGOS#(config content-filter) local
SGOS#(config iwf) download day-of-week all
 ok
SGOS#(config iwf) exit
SGOS#(config content-filter) exit
SGOS#(config)
```

#(config local)

Synopsis

Use this command to configure local content filtering.

Syntax

```
#(config) content-filter
```

This changes the prompt to:

```
#(config content-filter) local
```

This changes the prompt to:

```
#(config local)
```

Subcommands

`#(config local)` **`clear`**
Clears the local database from the system.

`#(config local)` **`download all-day`**
Checks for database updates all day.

`#(config local)` **`download auto`**
Enables automatic database downloads.

`#(config local)` **`download between-hours`** *`start stop`*
Sets the interval for automatic database update checks.

`#(config local)` **`download encrypted-password`** *`encrypted_password`*
Specifies the encrypted password for the database download server.

`#(config local)` **`download get-now`**
Initiates an immediate database download.

`#(config local)` **`download password`** *`password`*
Specifies the password for the database download server.

`#(config local)` **`download url`** {**`default`** | *`url`*}
Specifies using either the default URL or a specific URL for the database download server.

`#(config local)` **`download username`** *`username`*
Specifies the username for the database download server.

`#(config local)` **`exit`**
Exits configure local database mode and returns to configure content-filter mode.

`#(config local)` **`no download auto`**
Disables automatic download.

`#(config local)` **`no download encrypted-password`**
Clears the encrypted password for the database download server.

`#(config local)` **`no download password`**
Clears the password for the database download server.

`#(config local)` **`no download url`**
Clears the URL for the database download server.

`#(config local)` **`no download username`**
Clears the username for the database download server.

`#(config local) source`
Shows the database source file.

`#(config local) view`
Shows the current local database settings.

For More Information

- "Filtering Web Content" in *SGOS Administration Guide*

Example

```
SGOS#(config) content-filter
SGOS#(config content-filter) local
SGOS#(config local) download day-of-week all
 ok
SGOS#(config local) exit
SGOS#(config content-filter) exit
SGOS#(config)
```

#(config optenet)

Synopsis

Use this command to configure Optenet content filtering.

Syntax

```
#(config) content-filter
```

This changes the prompt to:

```
#(config content-filter) optenet
```

This changes the prompt to:

```
#(config optenet)
```

Subcommands

#(config optenet) download all-day
Checks for database updates all day.

#(config optenet) download auto
Enables automatic database downloads.

#(config optenet) download between-hours *start stop*
Sets the interval for automatic database update checks.

#(config optenet) download encrypted-password *encrypted_password*
Specifies the encrypted password for the database download server.

#(config optenet) download password *password*
Specifies the password for the database download server.

#(config optenet) download url {**default** | *url*}
Specifies using either the default URL or a specific URL for the database download server.

#(config optenet) download username *username*
Specifies the username for the database download server.

#(config optenet) exit
Exits configure optenet mode and returns to configure content-filter mode.

#(config optenet) no download auto
Disables automatic download.

#(config optenet) no download encrypted-password
Clears the encrypted password for the database download server.

#(config optenet) no download password
Clears the password for the database download server.

#(config optenet) no download url
Clears the URL for the database download server.

#(config optenet) no download username
Clears the username for the database download server.

#(config optenet) view
Shows the current optenet Web Filter settings.

For More Information

- "Filtering Web Content" in *SGOS Administration Guide*

Example

```
SGOS#(config) content-filter
SGOS#(config content-filter) optenet
SGOS#(config optenet) download time-of-day 20
 ok
SGOS#(config optenet) exit
SGOS#(config content-filter) exit
SGOS#(config)
```

#(config proventia)

Synopsis

Use this command to configure Proventia Web Filter content filtering.

Syntax

```
#(config) content-filter
```

This changes the prompt to:

```
#(config content-filter) proventia
```

This changes the prompt to:

```
#(config proventia)
```

Subcommands

`#(config proventia)` **`download all-day`**
Checks for database updates all day.

`#(config proventia)` **`download auto`**
Enables automatic database downloads.

`#(config proventia)` **`download between-hours`** *`start stop`*
Sets the interval for automatic database update checks.

`#(config proventia)` **`download encrypted-password`** *`encrypted_password`*
Specifies the encrypted password for the database download server.

`#(config proventia)` **`download get-now`**
Initiates an immediate database download.

`#(config proventia)` **`download password`** *`password`*
Specifies the password for the database download server.

`#(config proventia)` **`download url`** {**`default`** | *`url`*}
Specifies using either the default URL or a specific URL for the database download server.

`#(config proventia)` **`download username`** *`username`*
Specifies the username for the database download server.

`#(config proventia)` **`exit`**
Exits configure proventia mode and returns to configure content-filter mode.

`#(config proventia)` **`no download auto`**
Disables automatic download.

`#(config proventia)` **`no download encrypted-password`**
Clears the encrypted password for the database download server.

`#(config proventia)` **`no download password`**
Clears the password for the database download server.

`#(config proventia)` **`no download url`**
Clears the URL for the database download server.

`#(config proventia)` **`no download username`**
Clears the username for the database download server.

`#(config proventia)` **`view`**
Shows the current proventia Web Filter settings.

For More Information

- "Filtering Web Content" in *SGOS Administration Guide*

Example

```
SGOS#(config) content-filter
SGOS#(config content-filter) proventia
SGOS#(config proventia) download time-of-day 20
 ok
SGOS#(config proventia) exit
SGOS#(config content-filter) exit
SGOS#(config)
```

#(config surfcontrol)

Synopsis

Use this command to configure SurfControl filters that control the type of content retrieved by the ProxySG and filter requests made by clients.

Syntax

#(config) **content-filter**

This changes the prompt to:

```
#(config content-filter) surfcontrol
```

This changes the prompt to:

```
#(config surfcontrol)
```

Subcommands

`#(config surfcontrol)` **download all-day**
Checks for database updates all day.

`#(config surfcontrol)` **download auto**
Enables automatic database downloads.

`#(config surfcontrol)` **download between-hours** *start stop*
Sets the interval for automatic database update checks.

`#(config surfcontrol)` **encrypted-password** *encrypted-password*
Sets the download encrypted password. The username/password is assigned by Blue Coat.

`#(config surfcontrol)` **download get-now**
Initiates immediate database download. If a full download is unnecessary, an incremental download is initiated.

`#(config surfcontrol)` **download url** {**default** | *url*}
Specifies using either the default URL or a specific URL for the database download server.

`#(config surfcontrol)` **download username** *username*
Sets the download username. The username/password is assigned by Blue Coat.

`#(config surfcontrol)` **exit**
Exits configure surfcontrol mode and returns to configure content-filter mode

`#(config surfcontrol)` **no download** {**auto** | **encrypted-password** | **username** | **password** | **url**}
Negates download commands.

`#(config surfcontrol)` **view**
Shows the current SurfControl settings.

For More Information

- "Filtering Web Content" in *SGOS Administration Guide*

Example

```
SGOS#(config) content-filter
SGOS#(config content-filter) surfcontrol
SGOS#(config surfcontrol) no download url
 ok
SGOS#(config surfcontrol) exit
SGOS#(config content-filter) exit
SGOS#(config)
```

#(config webwasher)

Synopsis

Use this command to configure Webwasher URL Filter content filtering.

Syntax

```
#(config) content-filter
```

This changes the prompt to:

```
#(config content-filter) webwasher
```

This changes the prompt to:

```
#(config webwasher)
```

Subcommands

`#(config webwasher)` **`download all-day`**
Checks for database updates all day.

`#(config webwasher)` **`download auto`**
Enables automatic database downloads.

`#(config webwasher)` **`download between-hours`** *`start stop`*
Sets the interval for automatic database update checks.

`#(config webwasher)` **`download encrypted-password`** *`encrypted_password`*
Specifies the encrypted password for the database download server.

`#(config webwasher)` **`download get-now`**
Initiates an immediate database download. If a full download is unnecessary, an incremental download is initiated.

`#(config webwasher)` **`download password`** *`password`*
Specifies the password for the database download server.

`#(config webwasher)` **`download url`** {**`default`** | *`url`*}
Specifies using either the default URL or a specific URL for the database download server.

`#(config webwasher)` **`download username`** *`username`*
Specifies the username for the database download server.

`#(config webwasher)` **`exit`**
Exits configure webwasher mode and returns to configure content-filter mode.

`#(config webwasher)` **`no download auto`**
Disables automatic download.

`#(config webwasher)` **`no download encrypted-password`**
Clears the encrypted password for the database download server.

`#(config webwasher)` **`no download password`**
Clears the password for the database download server.

`#(config webwasher)` **`no download url`**
Clears the URL for the database download server.

`#(config webwasher)` **`no download username`**
Clears the username for the database download server.

`#(config webwasher)` **`view`**
Shows the current webwasher Web Filter settings.

For More Information

- "Filtering Web Content" in *SGOS Administration Guide*

Example

```
SGOS#(config) content-filter
SGOS#(config content-filter) webwasher
SGOS#(config webwasher) download time-of-day 20
 ok
SGOS#(config webwasher) exit
SGOS#(config content-filter) exit
SGOS#(config)
```

#(config) connection-forwarding

Synopsis

This command enables you to configure the TCP Connection Forwarding aspect of ADN transparent tunnel load balancing and asymmetric routing.

Syntax

```
#(config) connection-forwarding
```

This changes the prompt to:

```
#(config connection-forwarding)
```

Subcommands

`SGOS# (config connection forwarding) {add | remove} ip_address`
Add or remove a ProxySG to a connection forwarding peer group.

`SGOS# (config connection forwarding) port number`
Specify the port used by all peers in the peer group to communicate connection information (each peer in the group *must* use the same port number). The default is 3030.

`SGOS# (config connection forwarding) {enable | disable}`
Enables or disables connection forwarding on this ProxySG.

`SGOS# (config connection forwarding) clear`
Clear the list of forwarding peers from this ProxySG.

`SGOS# (config connection forwarding) exit`
Exits `(config connection forwarding)` mode and returns to `#(config)` mode.

`SGOS# (config connection forwarding) view`
View the TCP connection forwarding information.

For More Information

- *SGOS Administration Guide*

Example

```
SGOS#(config) connection-forwarding
SGOS#(connection-forwarding) add 10.9.59.100
 ok
SGOS#(config connection-forwarding) port 3030
 ok
SGOS#(config connection-forwarding) enable
 ok
```

#(config) diagnostics

Synopsis

This command enables you to configure the remote diagnostic feature Heartbeat.

Syntax

```
#(config) diagnostics
```

This changes the prompt to:

```
#(config diagnostics)
```

Subcommands

`#(config diagnostics)` **`cpu-monitor {disable | enable}`**
Enables or disables the CPU monitor (the CPU monitor is disabled by default).

`#(config diagnostics)` **`cpu-monitor interval`** *`seconds`*
Sets the periodic interval of the CPU monitor from 1 to 59 seconds (the default setting is 5 seconds).

`#(config diagnostics)` **`exit`**
Exits `#(config diagnostics)` mode and returns to `#(config)` mode.

`#(config diagnostics)` **`heartbeat {disable | enable}`**
Enables or disables the ProxySG Heartbeat features.

`#(config diagnostics)` **`monitor {disable | enable}`**
Enables or disables the Blue Coat monitoring feature.

`#(config diagnostics)` **`send-heartbeat`**
Triggers a heartbeat report.

`#(config diagnostics)` **`service-info`**
Changes the prompt (see **`#(config service-info)`** on page 179)

`#(config diagnostics)` **`snapshot (create | delete}`** *`snapshot_name`*
Create or delete a snapshot job. By default, the `sysinfo` snapshot job keeps the last 100 snapshots. The `sysinfo_stats` snapshot job keeps the last 168 snapshots. Snapshots created in SGOS 6.5.2 or later are not viewable if you downgrade to SGOS 6.5.1 or earlier.

`#(config diagnostics)` **`edit`** *`snapshot_name`*
Changes the prompt to **`#(config snapshot snapshot_name)`** on page 181)

`#(config diagnostics)` **`view configuration`**
Displays diagnostics settings for Heartbeats, CPU monitor, automatic service-info, and snapshots.

`#(config diagnostics)` **`view cpu-monitor`**
Displays the CPU Monitor results.

`#(config diagnostics)` **`view service-info`**
Displays service-info settings and progress.

`#(config diagnostics)` **`view snapshot`** *`snapshot_name`*
Displays the snapshot settings (target, status, interval, to keep, to take, and next snapshot) for the snapshot name specified.

For More Information

- *SGOS Administration Guide*

Example

```
SGOS#(config) diagnostics
SGOS#(config diagnostics) heartbeat enable
 ok
SGOS#(config diagnostics) exit
SGOS#(config)
```

#(config service-info)

Synopsis

This command allows you to send service information to Blue Coat.

Syntax

```
#(config) diagnostics
```

This changes the prompt to:

```
#(config diagnostics) service-info
```

This changes the prompt to:

```
#(config service-info)
```

Subcommands

`#(diagnostics service-info) auto {disable | enable}`
Disables or enables the automatic service information feature.

`#(diagnostics service-info) auto no sr-number`
Clears the service-request number for the automatic service information feature.

`#(diagnostics service-info) auto sr-number sr_number`
Sets the service-request number for the automatic service information feature.

`#(diagnostics service-info) bandwidth-class bandwidth class name`
Sets a bandwidth class used to manage the bandwidth of service-information transfers.

In order to do bandwidth-manage service-information transfers, bandwidth management must be enabled. You must also create a bandwidth class for service-information transfers (in bandwidth-management mode) before you can select it here.

`#(diagnostics service-info) cancel all`
Cancel all service information being sent to Blue Coat.

`#(diagnostics service-info) cancel one_or_more_from_view_status`
Cancel certain service information being sent to Blue Coat.

`#(diagnostics service-info) exit`
Exits `#(config diagnostics service-info)` mode and returns to `#(config diagnostics)` mode.

`#(diagnostics service-info) no bandwidth-class`
Disables bandwidth-management for service-information transfers

`#(diagnostics service-info) send sr_number one_or_more_commands_from_view_available`
Sends a specific service request number along with a specific command or commands (chosen from the list provided by the `view available` command) to Blue Coat.

`#(diagnostics service-info) view available`
Shows list of service information than can be sent to Blue Coat.

`#(diagnostics service-info) view status`
Shows transfer status of service information to Blue Coat.

For More Information

- **#(config) bandwidth-management** on page 135
- *SGOS Administration Guide*

Example

```
SGOS#(config) diagnostics
SGOS#(config diagnostics) service-info
SGOS#(diagnostics service-info) view available
Service information that can be sent to Blue Coat

Name                                    Approx Size (bytes)
Event_log                               188,416
System_information                      Unknown
Snapshot_sysinfo                        Unknown
Snapshot_sysinfo_stats                  Unknown
SGOS#(diagnostics service-info) send 1-4974446 event_log system_information
snapshot_sysinfo
Sending the following reports
Event_log
System_information
Snapshot_sysinfo
SGOS#(diagnostics service-info) view status
Name                                      Transferred
Event_log                                 Transferred successfully
Snapshot_sysinfo                          Transferred successfully
Event_log                                 Transferred successfully
System_information                        Transferred successfully
SGOS#(diagnostics service-info) exit
SGOS#(config diagnostics) exit
SGOS#(config)
```

#(config snapshot *snapshot_name)*

Synopsis

This command allows you to edit a snapshot job.

Syntax

```
#(config) diagnostics
```

This changes the prompt to:

```
#(config diagnostics) snapshot edit snapshot_name
```

This changes the prompt to:

```
#(config snapshot snapshot_name)
```

Subcommands

`#(config snapshot snapshot_name)` **`clear-reports`**
Clears all stored snapshots reports.

`#(config snapshot snapshot_name)` {**`disable`** | **`enable`**}
Disables or enables this snapshot job.

`#(config snapshot snapshot_name)` **`exit`**
Exits `#(config diagnostics snapshot_name)` mode and returns to `#(config diagnostics service-info)` mode.

`#(config snapshot snapshot_name)` **`interval`** *`minutes`*
Specifies the interval between snapshots reports in minutes.

`#(config snapshot snapshot_name)` **`keep`** *`number_to_keep (from 1 - 1000)`*
Specifies the number of snapshot reports to keep.

`#(config snapshot snapshot_name)` **`take`** {**`infinite`** | *`number_to_take`*}
Specifies the number of snapshot reports to take.

`#(config snapshot snapshot_name)` **`target`** *`object_to_fetch`*
Specifies the object to snapshot.

`#(config snapshot snapshot_name)` **`view`**
Displays snapshot status and configuration.

For More Information

- *SGOS Administration Guide*

Example

```
SGOS#(config) diagnostics
SGOS#(config diagnostics) snapshot testshot
SGOS#(diagnostics snapshot testshot) enable
 ok
SGOS#(diagnostics service-info) interval 1440
 ok
SGOS#(diagnostics snapshot testshot) exit
SGOS#(config diagnostics) exit
SGOS#(config)
```

#(config) dns

Synopsis

The `dns` command enables you to modify the DNS settings for the ProxySG. Note that the alternate DNS servers are only checked if the servers in the standard DNS list return: "Name not found."

Syntax

```
#(config) dns [subcommands]
```

Subcommands

`#(config) dns clear imputing`
Sets all entries in the name imputing list to null.

`#(config) dns client-affinity {disable | enable}`
Enable or disable client-affinity.
When enabled, requests from the same client resolve the hostname in the same order. www.google.com resolves to 66.102.7.99, 66.102.7.147, and 66.102.7.104. If client-affinity is enabled and the ProxySG receives a request (http, streaming or other proxy request) for www.google.com, it uses the client's IP address to determine the order of the resolved addresses. If client-affinity is disabled, the order of the resolved addresses changed each time the ProxySG receives a request.

`#(config) dns imputing name`
Identifies the file indicated by *name* as the name imputing list.

`#(config) dns negative-cache-ttl-override seconds`
Set the DNS negative cache time-to-live value for *seconds*.
A DNS request to an unknown domain name (klauwjdasd.bluecaot.com) is cached by the ProxySG. This type of caching is called a negative cache because it does not resolve to an actual IP address. The TTL value for a negative cache entry can be overwritten by this command.

`#(config) dns no imputing imputed_name`
Removes the imputed name identified by *imputed_name* from the name imputing list.

`#(config) dns no negative-cache-ttl-override`
Do not override the negative cache time-to-live value.

`#(config) dns recursion (disable | enable)`

Enable or disable DNS recursion. By default, recursion is disabled. When recursion is enabled, if a server returns authoritative server information instead of an A record, the ProxySG follows the referrals until it receives an answer or detects a recursion loop. If there are more than eight referrals, the ProxySG assumes that there is a loop and aborts the request.

For More Information

- *SGOS Administration Guide*

Example

```
SGOS#(config) dns imputing name bluecoat.com
 ok
SGOS#(config) dns clear imputing
 ok
SGOS#(config) dns recursion enable
 ok
```

#(config) dns-forwarding

Synopsis

The `dns-forwarding` command enables you to create, delete, and edit DNS forwarding groups for the ProxySG.

Syntax

`#(config)` **`dns-forwarding`**

This changes the prompt to:

`#(config dns forwarding)`

Subcommands

`#(config dns forwarding)` **`create`** *`group-alias`* [*`host-ip`*]
Creates a DNS forwarding group.

`#(config dns forwarding)` **`delete`** *`group-alias`*
Deletes a DNS forwarding group.

`#(config dns forwarding)` **`edit`** {**`primary`** | **`alternate`** | *`group-alias`*}
Edit a DNS forwarding group. Changes the prompt to **`#(config dns forwarding group_name)`** on page 185

`#(config dns forwarding)` **`exit`**
Exits `#(config dns forwarding)` mode and returns to `#(config)` mode.

`#(config dns forwarding)` **`view`**
Displays snapshot status and configuration.

For More Information

- *SGOS Administration Guide*

Examples

```
SGOS#(config dns forwarding) create testgroup 1.1.1.1
 ok
SGOS#(config dns forwarding) delete testgroup
 ok
SGOS#(config dns forwarding) edit primary
SGOS#(config dns forwarding primary) exit
SGOS#(config dns forwarding) view
DNS Forwarding configuration:
    Group: testgroup
      Servers:
        1.1.1.1
      Domains:
    Group:  primary
      Servers:
      Domains:
        *
    Group:  alternate
      Servers:
```

```
      Domains:
        *
SGOS#(config dns forwarding) exit
SGOS#(config)
```

#(config dns forwarding *group_name*)

Synopsis

This command allows you to edit a DNS fowarding group.

Syntax

```
#(config dns forwarding) edit {primary | alternate | group-alias}
```

This changes the prompt to:

```
#(config dns fowarding group_name)
```

Subcommands

`#(config dns fowarding group_name) add {domain domain | server server ip}`
Add domains or DNS servers to this group. IP addresses can be IPv4 or IPv6.

`#(config dns fowarding group_name) clear {domain | server}`
Clear the domain or server list for this group.

`#(config dns fowarding group_name) demote server_ip[slots]`
Demote the specified server IP address.

`#(config dns fowarding group_name) exit`
Return to the `#(config dns forwarding)` prompt.

`#(config dns fowarding group_name) promote server_ip[slots]`
Promote the specified server IP address in the DNS server list the number of places indicated. Must be a positive number. If the number is greater than the number of servers in the list, the server is promoted to the first entry in the list.

`#(config dns fowarding group_name) remove {domain | server}`
Remove a domain or server from the list.

`#(config dns fowarding group_name) view`
View the DNS forwarding configuration for this group.

For More Information

- *SGOS Administration Guide*

Examples

```
SGOS#(config dns forwarding primary) add server 1.1.1.1
 ok
SGOS#(config dns forwarding primary) demote 1.1.1.1
% Server is already last in the list.
SGOS#(config dns forwarding primary) promote 1.1.1.1
SGOS#(config dns forwarding primary) view
     Group:  primary
        Servers:
         1.1.1.1
         1.2.1.1
        Domains:
          *
SGOS#(config dns forwarding primary) exit
SGOS#(config dns forwarding)
```

#(config) event-log

Synopsis

You can configure the ProxySG to log system events as they occur. Event logging allows you to specify the types of system events logged, the size of the event log, and to configure Syslog monitoring. The ProxySG can also notify you by e-mail if an event is logged.

Syntax

```
#(config) event-log
```

This changes the prompt to:

```
#(config event-log)
```

Subcommands

`#(config event-log) exit`
Exits `#(config event-log)` mode and returns to `#(config)` mode.

`#(config event-log) level configuration`
Writes severe and configuration change error messages to the event log.

`#(config event-log) level informational`
Writes severe, configuration change, policy event, and information error messages to the event log.

`#(config event-log) level policy`
Writes severe, configuration change, and policy event error messages to the event log.

`#(config event-log) level severe`
Writes only severe error messages to the event log.

`#(config event-log) level verbose`
Writes all error messages to the event log.

`#(config event-log) log-size` *megabytes*
Specifies the maximum size of the event log in megabytes.

`#(config event-log) mail add` *email_address*
Specifies an e-mail recipient for the event log output.

`#(config event-log) mail clear`
Removes all e-mail recipients from the event log e-mail output distribution list.

`#(config event-log) mail no smtp-gateway`
Clears the SMTP gateway used for notifications. This command has been deprecated; use the **smtp** command instead. See **#(config) smtp** on page 383.

`#(config event-log) mail remove` *email_address*
Removes the e-mail recipient indicated by *email_address* from the event log e-mail output distribution list.

`#(config event-log) mail smtp-gateway {`*domain_name* | *ip_address*`}`
Specifies the SMTP gateway to use for event log e-mail output notifications. This command has been deprecated; use the **smtp** command instead. See **#(config) smtp** on page 383.

`#(config event-log) mail from` *from_address*
Specifies the 'From:' email address field for notifications. This command has been deprecated; use the **smtp** command instead. See **#(config) smtp** on page 383.

`#(config event-log) syslog add {`*host_name* | *ip_address*`}`
Adds a system logging loghost. Enter the IPv4 or IPv6 address of your loghost server, or specify a domain name that resolves to an IPv4 or IPv6 address.

`#(config event-log)` **`syslog clear`**
Removes all loghosts from system logging notification.

`#(config event-log)` **`syslog`** {**`disable`** | **`enable`**}
Disables or enables system logging notifications.

`#(config event-log)` **`syslog facility`** {**`auth`** | **`daemon`** | **`kernel`** | **`local0`** | **`local1`** | **`local2`** | **`local3`** | **`local4`** | **`local5`** | **`local6`** | **`local7`** | **`lpr`** | **`mail`** | **`news`** | **`syslog`** | **`user`** | **`uucp`**}
Sets the facility that is used when sending to a syslog server.

`#(config event-log)` **`syslog remove`** {*`host_name`* | *`ip_address`*}
Removes the specified system logging loghost.

`#(config event-log)` **`view`** [**`configuration`**] [**`start`** [**`YYYY-mm-dd`**] [**`HH:MM:SS`**]] [**`end`** [**`YYYY-mm-dd`**] [**`HH:MM:SS`**]] [**`regex`** *`regex`* | **`substring`** *`string`*]
View the event-log configuration, using the #(config event-log) configuration command, or view the contents of the event-log, using the filters offered to narrow the view.

`#(config event-log)` **`when-full`** {**`overwrite`** | **`stop`**}
Specifies what should happen to the event log when the maximum size has been reached. overwrite overwrites the oldest information in a FIFO manner; stop disables event logging.

For More Information

- *SGOS Administration Guide*

Example

```
SGOS#(config) event-log
SGOS#(config event-log) syslog enable
 ok
```

#(config) exceptions

Synopsis

These commands allow you to configure built-in and user-defined exception response objects.

Syntax

```
#(config) exceptions
```

This changes the prompt to:

```
#(config exceptions)
```

Subcommands

`#(config exceptions)` **`create`** *`exception_id`*
Creates the given exception.

`#(config exceptions)` **`company-name`** *`name`*
Sets the name used for the $(exception.company_name) substitution.

`#(config exceptions)` **`delete`** *`exception_id`*
Deletes the exception specified by *`exception_id`*.

`#(config exceptions)` **`edit`** *`exception_id or user_defined_exception_id`*
Changes the prompt to **`#(config exceptions [user-defined.]exception_id)`** on page 190.

`#(config exceptions)` **`exit`**
Exits `#(config exceptions)` mode and returns to `#(config)` mode.

`#(config exceptions)` **`http-code`**
E

`#(config exceptions)` **`inline`** {**`contact`** {*`eof_marker`*} | **`details`** {*`eof_marker`*} | **`format`** {*`eof_marker`*} | **`help`** {*`eof_marker`*} | **`http`** {**`contact`** {*`eof_marker`*} | **`details`** {*`eof_marker`*} | **`format`** {*`eof_marker`*} | **`help`** {*`eof_marker`*} | **`summary`** {*`eof_marker`*}} | **`summary`** {*`eof_marker`*}}
Configures defaults for all exception objects.

`#(config exceptions)` **`load exceptions`**
Downloads new exceptions.

`#(config exceptions)` **`no path`**
Clears the network path to download exceptions.

`#(config exceptions)` **`path`** *`url`*
Specifies the network path to download exceptions.

`#(config exceptions)` **`user-defined`** {**`inline`** {**`contact`** *`eof_marker`* | **`details`** *`eof_marker`* | **`format`** *`eof_marker`* | **`help`** *`eof_marker`* | **`http`** {**`contact`** *`eof_marker`* | **`details`** *`eof_marker`* | **`format`** *`eof_marker`* | **`help`** *`eof_marker`* | **`summary`** *`eof_marker`*} | **`summary`**} *`eof_marker`*} | **`http-code`** *`numeric http response code`*}
Configures the top-level values for user-defined exceptions.

For More Information

- *SGOS Administration Guide*

Example

```
SGOS#(config) exceptions
SGOS#(config exceptions) default contact
 ok
SGOS#(config exceptions) exit
SGOS#(config)
```

#(config exceptions [user-defined.]*exception_id)*

Synopsis

These commands allow you to edit an exception or a user-defined exception.

Syntax

```
#(config) exceptions
```

This changes the prompt to:

```
#(config exceptions) user_defined_exception_id
```

This changes the prompt to:

```
#(config exceptions user_defined_exception_id)
```

Subcommands

`#(config exceptions user-defined.`*`exception_id`*`) exit`
Exits `#(config exceptions user-defined.`*`exception_id`*`)` mode and returns to `#(config exceptions)` mode.

`#(config exceptions user-defined.`*`exception_id`*`) http-code` *`numeric_http_response_code`*
Configures this exception's HTTP response code.

`#(config exceptions user-defined.`*`exception_id`*`) inline {contact` *`eof_marker`* `| details` *`eof_marker`* `| format` *`eof_marker`* `| help` *`eof_marker`* `| http {contact` *`eof_marker`* `| details` *`eof_marker`* `| format` *`eof_marker`* `| help` *`eof_marker`* `| summary` *`eof_marker`*`} | summary` *`eof_marker`*`}`
Configures this exception's substitution values.

For More Information

- *SGOS Administration Guide*

Example

```
SGOS#(config) exceptions
SGOS#(config exceptions) edit testname
SGOS#(config exceptions user-defined.testname) http-code 000
 ok
SGOS#(config exceptions user-defined.testname) exit
SGOS#(config exceptions) exit
SGOS#(config)
```

#(config) exit

Synopsis

Exits from Configuration mode to Privileged mode, from Privileged mode to Standard mode. From Standard mode, the `exit` command closes the CLI session.

Syntax

```
#(config) exit
```

The `exit` command has no parameters or subcommands.

#(config) external-services

Synopsis

These commands allow you to configure your external services.

Use the edit ICAP commands to configure the ICAP service used to integrate the ProxySG with a virus scanning server. The configuration is specific to the virus scanning server and includes the server IP address, as well as the supported number of connections. If you are using the ProxySG with multiple virus scanning servers or multiple scanning services on the same server, add an ICAP service for each server or scanning service.

Note: When you define virus scanning policies, use the same service name. Make sure you type the ICAP service name accurately, whether you are configuring the service on the ProxySG or defining policies, since the name retrieves the other configuration settings for that service.

Syntax

```
#(config) external-services
```

This changes the prompt to:

```
#(config external-services)
```

Subcommands

`#(config external-services)` **`create icap`** *`icap_service_name`*
Creates an ICAP service.

`#(config external-services)` **`create service-group`** *`service_group_name`*
Creates a service group.

`#(config external-services)` **`delete`** *`name`*
Deletes an external service.

`#(config external-services)` **`edit`**
Changes the prompt to one of three external service edit commands:

`#(config icap icap_service_name)` on page 194

`#(config service-group service_group_name)` on page 196

`#(config external-services)` **`exit`**
Exits `#(config external-services)` mode and returns to `#(config)` mode.

`#(config external-services)` **`icap feedback interactive patience-page`** *`{seconds}`*
For traffic associated with a Web browser, display a patience page after the specified duration.

`#(config external-services)` **`icap feedback`** {**`interactive`** {**`trickle-start`** {*`seconds`*} | **`trickle-end`** {*`seconds`*} | **`none`**} | **`non-interactive`** {**`trickle-start`** {*`seconds`*} | **`trickle-end`** {*`seconds`*} | **`none`**}}
For interactive traffic (associated with a Web browser) or non-traffic (originating from a client other than a Web browser), employ a data trickling method so the user receives a small amount (trickle-start) or large amount (trickle-end) of object data while waiting for the results of the content scan (ICAP). Begin trickling after the specified duration.

`#(config external-services)` **`inline http icap-patience`** {**`details`** *`eof`* | **`header`** *`eof`* | **`help`** *`eof`* | **`summary`** *`eof`*}

Customizes ICAP patience page details for HTTP connections.

`#(config external-services)` **`inline ftp icap-patience text`** *`eof`*
Customizes ICAP patience page details for FTP connections.

`#(config external-services)` **`view`**
Shows external services and external service groups.

For More Information

- *SGOS Administration Guide*

Example

```
SGOS#(config) external-services
SGOS#(config external-services) create icap testicap
 ok
SGOS#(config external-services) exit
SGOS#(config)
```

#(config icap *icap_service_name)*

Synopsis

These commands allow you to edit ICAP parameters.

Syntax

```
#(config) external-services
```

This changes the prompt to:

```
#(config external-services) create icap icap_service_name
#(config external-services) edit icap_service_name
```

This changes the prompt to:

```
#(config icap icap_service_name)
```

Subcommands

`#(config icap icap_service_name) defer-threshold defer-threshold`
Sets the deferred scanning threshold.

`#(config icap icap_service_name) exit`
Exits `#(config ICAP name)` mode and returns to `#(config external-services)` mode.

`#(config icap icap_service_name) event-log connection-failure`
Enables event log options

`#(config icap icap_service_name) max-conn max_num_connections`
Sets the maximum number of connections for the ICAP service.

`#(config icap icap_service_name) methods {REQMOD | RESPMOD}`
Sets the method supported by the ICAP service. REQMOD is request modification and RESPMOD is response modification.

`#(config icap icap_service_name) no defer-threshold`
Disables the deferred scanning threshold.

`#(config icap icap_service_name) no event-log connection-failure`
Disables event log options

`#(config icap icap_service_name) no send {client-address | server-address}`
Specifies what should not be sent to the ICAP server.

`#(config icap icap_service_name) no notify virus-detected`
Specifies no notification to the administrator when a virus is detected.

`#(config icap icap_service_name) no port {port | default}`
Disables ports for both plain and secure ICAP.

`#(config icap icap_service_name) no preview`
Specifies that previews do not get sent.

`#(config icap icap_service_name) no secure-port`
Disables the secure ICAP mode.

`#(config icap icap_service_name) no ssl-device-profile ssl-device-profile`
Removes the selected SSL device profile.

`#(config icap icap_service_name) no use-vendor-virus-page`
Does not use the ProxySG's virus detected exception.

`#(config icap icap_service_name) notify virus-detected`
Specifies notification when viruses are found.

`#(config icap` *`icap_service_name`*`)` **`port`** {*`port`* | **`default`**}
Sets the plain ICAP port. Enter the desired port or the default port. The default port is 1344. To enter another port, enter a value from 1– 65534.

`#(config icap` *`icap_service_name`*`)` **`preview-size`** *`bytes`*
Sets the preview size for the ICAP service.

`#(config icap` *`icap_service_name`*`)` **`secure-port`** {*`port`* | **`default`**}
Sets the secure ICAP port. Enter the desired port or the default port. The default port is 11344. To enter another port, enter a value from 1– 65534. This command can only be used if an SSL device profile is not specified.

`#(config icap` *`icap_service_name`*`)` **`send client-address`**
Specifies that the client address be sent to the ICAP service.

`#(config icap` *`icap_service_name`*`)` **`send server-address`**
Specifies that the server address be sent to the ICAP service.

`#(config icap` *`icap_service_name`*`)` **`send authenticated-groups`**
Specifies that the authenticated groups be sent to the ICAP service.

`#(config icap` *`icap_service_name`*`)` **`send authenticated-user`**
Specifies that the authenticated user be sent to the ICAP service.

`#(config icap` *`icap_service_name`*`)` **`sense-settings`**
Senses the service's setting by contacting the server.

`#(config icap` *`icap_service_name`*`)` **`ssl-device-profile`** *`ssl-device-profile`*
Associates an SSL device profile with the ICAP service. No device profile is the default.

`#(config icap` *`icap_service_name`*`)` **`timeout`** *`seconds`*
Sets the connection timeout for the ICAP services.

`#(config icap` *`icap_service_name`*`)` **`url`** *`url`*
Sets the URL for the ICAP services.

`#(config icap` *`icap_service_name`*`)` **`use-vendor-virus-page`**
Use the ICAP vendor's virus detected page.

`#(config icap` *`icap_service_name`*`)` **`view`**
Displays the service's current configuration.

For More Information

- *SGOS Administration Guide*

Example

```
SGOS#(config) external-services
SGOS#(config external-services) edit testicap
SGOS#(config icap testicap) send client-address
 ok
SGOS#(config icap testicap) exit
SGOS#(config external-services) exit
SGOS#(config)
```

#(config service-group *service_group_name)*

Synopsis

These commands allow you to edit service group parameters.

Syntax

`#(config)` **`external-services`**

This changes the prompt to:

`#(config external-services)` **`create service-group`** *`service_group_name`*

`#(config external-services)` **`edit`** *`service_group_name`*

This changes the prompt to:

`#(config service-group` *`service_group_name`*`)`

Subcommands

`#(config service-group` *`service_group_name`*`)` **`add`** *`entry_name`*
Adds an entry to this service group.

`#(config service-group` *`service_group_name`***`) edit`** *`entry_name`*
Changes the prompt to `#(config service-group` *`service_group_name entry_name`*`).`

`#(config service-group` *`service_group_name entry_name`*`)` **`exit`**
Exits `#(config service-group` *`name/entry`* `name)` mode and returns to `#(config service-group` *`name`*`)` mode.

`#(config service-group` *`service_group_name entry_name`*`)` **`view`**
Shows this entry's configuration.

`#(config service-group` *`service_group_name entry_name`*`)` **`weight`** ***`0 to 255`***
Modifies this entry's weight.

`#(config service-group` *`service_group_name`*`)` **`exit`**
Exits `#(config service-group_`*`name`*`)` mode and returns to `#(config external-services)` mode.

`#(config service-group` *`service_group_name`*`)` **`remove`** *`entry_name`*
Removes an entry from this service group.

`#(config service-group` *`service_group_name`*`)` **`view`**
Displays this service group's configuration.

For More Information

- *SGOS Administration Guide*

Examples

```
SGOS#(config) external-services
SGOS#(config external-services) edit testgroup
SGOS#(config service-group testgroup) add testentry
 ok
SGOS#(config service-group testgroup) exit
SGOS#(config external-services) exit
SGOS#(config)
```

```
SGOS#(config) external-services
SGOS#(config external-services) edit testgroup
SGOS#(config service-group testgroup) edit testentry
SGOS#(config service-group testgroup testentry) weight 223
 ok
SGOS#(config service-group testgroup testentry) exit
SGOS#(config service-group testgroup) exit
SGOS#(config external-services) exit
SGOS#(config)
```

#(config) failover

Synopsis

These commands allow you to configure redundancy into your network.

Syntax

```
#(config) failover
```

This changes the prompt to:

```
#(config failover)
```

Subcommands

`#(config failover)` **`create`** *`group_address`*
Creates a failover group.

`#(config failover)` **`delete`** *`group_address`*
Deletes a failover group.

`#(config failover)` **`edit`** *`group_address`*
Changes the prompt to `#(config failover` *`group_address`*).

`#(config failover` *`group_address`*`)` {**`disable`** | **`enable`**}
Disables or enables failover group indicated by *`group_address.`*

`#(config failover` *`group_address`*`)` **`encrypted-secret`** *`encrypted_secret`*
(Optional but recommended) Refers to an encrypted password shared only with the group.

`#(config failover` *`group_address`*`)` **`exit`**
Exits `#(config failover` *`group_address`*`)` mode and returns to `#(config failover)` mode.

`#(config failover` *`group_address`*`)` **`interval`** *`interval_in_seconds`*
(Optional) Refers to the time between advertisements from the master to the multicast address. The default is 40 seconds.

`#(config failover` *`group_address`*`)` **`master`**
Defines the current system as the master and all other systems as slaves.

`#(config failover` *`group_address`*`)` **`multicast-address`** *`multicast_address`*
Refers to a multicast address where the master sends the keepalives (advertisements) to the slave systems.

`#(config failover` *`group_address`*`)` **`no interval`**
Resets the interval to the default value (40 seconds).

`#(config failover` *`group_address`*`)` **`no multicast-address`**
Removes the multicast address from the failover group.

`#(config failover` *`group_address`*`)` **`no master`**
Removes as configured master.

`#(config failover` *`group_address`*`)` **`no priority`**
Resets the priority to the default value (100).

`#(config failover` *`group_address`*`)` **`no secret`**
Clears the secret from the failover group.

`#(config failover` *`group_address`*`)` **`priority`** *`relative_priority`*
(Optional) Refers to the rank of slave systems. The range is from 1 to 253. (The master system, the one whose IP address matches the group address, gets 254.)

`#(config failover` *group_address*`)` **`secret`** *secret*
(Optional but recommended) Refers to a password shared only with the group. You can create a secret, which is then hashed.

`#(config failover` *group_address*`)` **`view`**
Shows the current settings for the failover group indicated by *group_address*.

`#(config failover)` **`exit`**
Exits `#(config failover)` mode and returns to `#(config)` mode.

`#(config failover)` **`view`** {**`configuration`** [*group_address* | **`<Enter>`**] | **`statistics`**}
View the configuration of a group or all groups or view all statistics.

For More Information

- *SGOS Administration Guide*

Examples

```
SGOS#(config) failover
SGOS#(config failover) create 10.9.17.135
 ok
SGOS#(config failover) exit
SGOS#(config)
```

```
SGOS#(config) failover
SGOS#(config failover) edit 10.9.17.135
SGOS#(config failover 10.9.17.135) master
 ok
SGOS#(config failover 10.9.17.135) exit
SGOS#(config failover) exit
```

#(config) forwarding

Synopsis

Configures forwarding of content requests to defined hosts and groups through policy.

Syntax

```
#(config) forwarding
```

This changes the prompt to:

```
#(config forwarding)
```

Subcommands

`#(config forwarding)` **`create host`** *`host_alias host_name`* `[http[=port] [https[=port]] [ftp[=port]] [mms[=port]] [rtsp[=port]] [tcp[=port]] [telnet[=port]] [ssl-verify-server[=yes | =no]] [group=group_name] [server | proxy]`

The forwarding host (*`host_name`*) can be an IPv4 or IPv6 host or address.

`#(config forwarding)` **`create group`** *`group_name`*
Creates a forwarding host/group. The only required entries under the `create` option (for a host) are *`host_alias`*, *`host_name`*, a protocol, and a port number. The port number can be defined explicitly (i.e., `http=8080`), or it can take on the default port value of the protocol, if one exists (i.e., enter `http`, and the default port value of `80` is entered automatically).

To create a host group, you must also include the `group=`*`group_name`* command. If this is the first mention of the group, *`group_name`*, then that group is automatically created with this host as its first member. Do not use this command when creating an independent host.

`#(config forwarding)` **`default-sequence add`** *`host_or_group_alias`*
Adds an alias to the end of the default failover sequence.

`#(config forwarding)` **`default-sequence clear`**
Clears the default failover sequence.

`#(config forwarding)` **`default-sequence demote`** *`host_or_group_alias`*
Demotes an alias one place toward the end of the default failover sequence.

`#(config forwarding)` **`default-sequence promote`** *`host_or_group_alias`*
Promotes an alias one place toward the start of the default failover sequence.

`#(config forwarding)` **`default-sequence remove`** *`host_or_group_alias`*
Removes an alias from the default failover sequence.

`#(config forwarding)` **`delete all`**
Deletes all forwarding hosts and groups.

`#(config forwarding)` **`delete group`** *`group_name`*
Deletes only the group identified by *`group_name`*.

`#(config forwarding)` **`delete host`** *`host_alias`*
Deletes only the host identified by *`host_alias`*.

`#(config forwarding)` **`download-via-forwarding {disable | enable}`**
Disables or enables configuration file downloading using forwarding.

`#(config forwarding)` **`edit`** *`host_or_group_alias`*
Changes the prompt to:

- **`#(config forwarding group_alias)`** on page 203
- **`#(config forwarding host_alias)`** on page 205

`#(config forwarding)` **`exit`**
Exits `#(config forwarding)` mode and returns to `#(config)` mode.

`#(config forwarding)` **`failure-mode`** {**`closed`** | **`open`**}
Sets the default forwarding failure mode to closed or open.

`#(config forwarding)` **`host-affinity http method`** {**`accelerator-cookie`** [*`host_or_group_alias`*] | **`client-ip-address`** [*`host_or_group_alias`*] | **`default`** [*`host_or_group_alias`*] | **`none`** [*`host_or_group_alias`*]}
Selects a host affinity method for HTTP. If a host or group alias is not specified for the `accelerator-cookie`, `client-ip-address`, or `none` options, the global default is used. Use the `default` option to specify default configurations for all the settings for a specified host or group.

`#(config forwarding)` **`host-affinity ssl method`** {**`accelerator-cookie`** [*`host_or_group_alias`*] | **`client-ip-address`** [*`host_or_group_alias`*] | **`default`** [*`host_or_group_alias`*] | **`none`** [*`host_or_group_alias`*] | **`ssl-session-id`** [*`host_or_group_alias`*]}
Selects a host affinity method for SSL. If a host or group alias is not specified for the `accelerator-cookie`, `client-ip-address`, `none`, or `ssl-session-id` options, the global default is used. Use the `default` option to specify default configurations for all the settings for a specified host or group.

`#(config forwarding)` **`host-affinity other method`** {**`client-ip-address`** [*`host_or_group_alias`*] | **`default`** [*`host_or_group_alias`*] | **`none`** [*`host_or_group_alias`*]}
Selects a host affinity method (non-HTTP or non-SSL). If a host or group alias is not specified for the `client-ip-address`, or `none` options, the global default is used. Use the `default` option to specify default configurations for all the settings for a specified host or group.

`#(config forwarding)` **`host-affinity timeout`** *`minutes`*
Sets the timeout in minutes for the host affinity.

`#(config forwarding)` **`integrated-host-timeout`** *`minutes`*
Sets the timeout for aging out unused integrated hosts.

`#(config forwarding)` **`load-balance`** {**`default`** [*`group_alias`*] | **`domain-hash`** [*`group_alias`*] | **`least-connections`** [*`group_alias`*] | **`none`** [*`group_alias`*] | **`round-robin`** [*`group_alias`*] | **`url`** [*`group_alias`*]}
Sets if and how load balancing hashes between group members. If a group alias is not specified for the `domain-hash`, `least-connections`, `round-robin`, `url`, or `none` options, the global default is used. Use the `default` option to specify default configurations for all the settings for a specified group.

`#(config forwarding)` **`load-balance method`** {**`default`** [*`host_alias`*] | **`least-connections`** [*`host_alias`*] | **`none`** [*`host_alias`*] | **`round-robin`** [*`host_alias`*]}
Sets the load balancing method. If a host alias is not specified for the `least-connections`, `round-robin`, or `none` options, the global default is used. Use the `default` option to specify default configurations for all the settings for a specified host.

`#(config forwarding)` **`no path`**
Negates certain forwarding settings.

`#(config forwarding)` **`path`** *`url`*
Sets the network path to download forwarding settings.

`#(config forwarding)` **`view`**
Displays the currently defined forwarding groups or hosts.

For More Information

- *SGOS Administration Guide*

Example

```
SGOS#(config) forwarding
SGOS#(config forwarding) download-via-forwarding disable
 ok
SGOS#(config forwarding) failure-mode closed
 ok
SGOS#(config forwarding) host-affinity method client-ip-address
 ok
SGOS#(config forwarding) load-balance hash domain group_name1
 ok
SGOS#(config forwarding) exit
SGOS#(config)
```

#(config forwarding *group_alias)*

Synopsis

These commands allow you to edit the settings of a specific forwarding group.

Syntax

```
#(config) forwarding
```

This changes the prompt to:

```
#(config forwarding) create host_alias hostname protocol=port group=group_alias
#(config forwarding) edit group_alias
```

This changes the prompt to:

```
#(config forwarding group_alias)
```

Subcommands

`#(config forwarding group_alias) add`
Adds a new group.

`#(config forwarding group_alias) exit`
Exits `#(config forwarding group_alias)` mode and returns to `#(config forwarding)` mode.

`#(config forwarding group_alias) host-affinity http {accelerator-cookie | client-ip-address | default | none}`
Changes the host affinity method (non-SSL) for this group.

`#(config forwarding group_alias) host-affinity other {client-ip-address | default | none}`
Changes the other host affinity method for this group.

`#(config forwarding group_alias) host-affinity ssl {accelerator-cookie | client-ip-address | default | ssl-session-id | none}`
Changes the host affinity method (SSL) for this group.

`#(config forwarding group_alias) load-balance method {default | domain-hash | least-connections | none | round-robin | url-hash}`
Changes the load balancing method.

`#(config forwarding group_alias) remove`
Removes an existing group.

`#(config forwarding group_alias) view`
Shows the current settings for this forwarding group.

For More Information

- *SGOS Administration Guide*

Example

```
SGOS#(config) forwarding
SGOS#(config forwarding) edit test_group
SGOS#(config forwarding test_group) load-balance hash domain
 ok
SGOS#(config forwarding test_group) exit
SGOS#(config forwarding) exit
SGOS#(config)
```

#(config forwarding *host_alias)*

Synopsis

These commands allow you to edit the settings of a specific forwarding host.

Syntax

```
#(config) forwarding
```

This changes the prompt to:

```
#(config forwarding) create host_alias hostname protocol=port
#(config forwarding) edit host_alias
```

This changes the prompt to:

```
#(config forwarding host_alias)
```

Subcommands

`#(config forwarding host_alias) exit`
Exits `#(config forwarding host_alias)` mode and returns to `#(config forwarding)` mode.

`#(config forwarding host_alias) ftp [port]`
Changes the FTP port to the default port or to a port that you specify.

`#(config forwarding host_alias) host host_name`
Changes the host name.

`#(config forwarding host_alias) host-affinity http {accelerator-cookie | client-ip-address | default | none}`
Changes the host affinity method (non-SSL) for this host.

`#(config forwarding host_alias) host-affinity other {client-ip-address | default | none}`
Changes the other host affinity method for this host.

`#(config forwarding host_alias) host-affinity ssl {accelerator-cookie | client-ip-address | default | ssl-session-id | none}`
Changes the host affinity method (SSL) for this host.

`#(config forwarding host_alias) http [port]`
Changes the HTTP port to the default port or to a port that you specify.

`#(config forwarding host_alias) https [port]`
Changes the HTTPS port to the default port or to a port that you specify.

`#(config forwarding host_alias) load-balance method {default | least-connections | round-robin | none}`
Changes the load balancing method.

`#(config forwarding host_alias) mms [port]`
Changes the MMS port to the default port or to a port that you specify.

`#(config forwarding host_alias) no {ftp | http | https | mms | rtsp | ssl-verify-server | tcp | telnet}`
Deletes a setting for this host.

`#(config forwarding host_alias) proxy`
Makes the host a proxy instead of a server; any HTTPS or TCP ports are deleted.

`#(config forwarding host_alias) rtsp [port]`
Changes the RTSP port to the default port or to a port that you specify.

`#(config forwarding` *`host_alias`*`)` **`server`**
Makes the host a server instead of a proxy.

`#(config forwarding` *`host_alias`*`)` **`ssl-verify-server`**
Sets SSL to verify server certificates.

`#(config forwarding` *`host_alias`*`)` **`tcp`** `[`*`port`*`]`
Changes the TCP port to the default port or to a port that you specify.

`#(config forwarding` *`host_alias`*`)` **`telnet`** `[`*`port`*`]`
Changes the Telnet port to the default port or to a port that you specify.

`#(config forwarding` *`host_alias`*`)` **`view`**
Shows the current settings for this forwarding host.

For More Information

- *SGOS Administration Guide*

Example

```
SGOS#(config) forwarding
SGOS#(config forwarding) edit test_host
SGOS#(config forwarding test_host) server
 ok
SGOS#(config forwarding test_host) exit
SGOS#(config forwarding) exit
```

#(config) front-panel

Synopsis

Use this command to configure the front panel. For instance, the front-panel LCD behavior can be configured using the `backlight` command.

Syntax

```
#(config) front-panel
```

This changes the prompt to:

```
#(config front-panel)
```

Subcommands

`#(config front-panel)` **`backlight flash`**
The front-panel LCD is configured to flash, which can, for instance, help you locate a particular appliance in a room full of appliances.

`#(config front-panel)` **`backlight state {off | on | timeout}`**
The front-panel LCD is configured to be always turned on, always turned off, or to turn off after a specified length of time (use the backlight timeout command to configure the length of time).

`#(config front-panel)` **`backlight timeout`** *`seconds`*
Configures the length of time before the front-panel LCD turns off. You must also set the `backlight state timeout` command to configure timeout mode.

`#(config front-panel)` **`exit`**
Exits `#(config front-panel)` mode and returns to `#(config)` mode.

`#(config front-panel)` **`no backlight flash`**
Stops the front-panel LCD from flashing.

`#(config front-panel)` **`view`**
Displays the front panel settings.

For More Information

- *SGOS Administration Guide*

Example

```
SGOS#(config) front-panel
SGOS#(config front-panel) backlight state timeout
 ok
SGOS#(config front-panel) backlight timeout 60
 ok
SGOS#(config front-panel) exit
SGOS#(config)
```

#(config) ftp

Synopsis

Use this command to configure FTP parameters.

Syntax

`#(config) ftp login-syntax {raptor | checkpoint}`
Toggles between Raptor and Checkpoint login syntax. The default is Raptor.

Note: Neither proxy authentication for transparent FTP nor proxy chaining are supported with the Checkpoint syntax. When native FTP traffic from an FTP client (such as WSFtp) is being authenticated by the ProxySG using the Raptor syntax, the recommended authentication mode is `auto` or `proxy`.

`#(config) ftp no welcome-banner`
No text is displayed to an FTP client when a connection occurs.

`#(config) ftp passive-mode {enable | disable}`
Enables or disables support for passive mode to clients. This applies to allowing "PASV" method when IPv4 is in use, and applies to allowing "EPSV" method when IPv6 is in use.

`#(config) ftp welcome-banner` *banner*
Customizes the text displayed to an FTP client when a connection occurs.

For More Information

- *SGOS Administration Guide*
- **#(config caching ftp)** on page 145

Example

```
SGOS #(config) ftp login-syntax checkpoint
ok
```

#(config) general

Synopsis

Use these commands to set global defaults for user behavior when license limits are exceeded and trusting client-provided destination IP addresses.

Syntax

`SGOS#(config) general`

This changes the prompt to:

`SGOS#(config general)`

Subcommands

`SGOS#(config general) exit`
Returns to `#(config)` prompt.

`SGOS#(config general) reflect-client-ip {disable| enable}`
Configures the client IP reflection.

`SGOS#(config general) resource-overflow-action {bypass | drop}`
Configures the resource overflow action by choosing to either bypass or drop new connections when resources are scarce.

`SGOS#(config general) trust-destination-ip {enable | disable}`
Allows the ProxySG appliance to trust a client-provided destination IP address and not do a DNS lookup.

- Proxy Edition default: `disable`
- MACH5 Edition default: `enable`

`SGOS#(config general) user-overflow-action {bypass | none | queue}`
Set overflow behavior when there are more licensed-user connections going through the system than is allowed by the model license. The default is `none`.

`SGOS#(config general) view`
View general mode settings.

For More Information

- *SGOS Administration Guide*

Example

```
SGOS#(config general) trust-destination-ip enable
ok
```

#(config)geolocation

Synopsis

This command allows you to specify download parameters, disable geolocation settings, and view current geolocation settings.

Syntax

```
#(config) geolocation
```

This enters geolocation mode and changes the prompt to:

```
#(config geolocation)
```

Subcommands

```
#(config geolocation) download get-now
```

Downloads a new version of the geolocation database.

```
#(config geolocation) service disable
```

Disables the geolocation service.

```
#(config geolocation) service enable
```

Enables the geolocation service.

```
#(config geolocation) view
```

Displays geolocation statistics, such as license information, registration status, the download URL for the geolocation database, results of the last download, and the last successful download. This subcommand produces the same output as the **#show geolocation** command.

```
#(config geolocation) view countries
```

Displays the list of countries defined in the geolocation database (if one has been downloaded). This subcommand produces the same output as **#show geolocation countries**.

For More Information

- *SGOS Administration Guide*

Example

```
#(config geolocation)view
License Type:           Subscription
Licensed Until:         Thu, 01 Jan 2015 00:00:00 UTC
Service:                Enabled
Download method:        Direct
Last successful download:
  Time:                 Wed, 10 Apr 2013 17:16:54 UTC
  Downloading from:     https://subscription.es.bluecoat.com/geoip/database
  Version:              20130402
#(config geolocation)
```

#(config) health-check

Synopsis

Use this command to configure health check settings.

Syntax

```
#(config) health-check
```

This changes the prompt to:

```
#(config health-check)
```

Subcommands

`#(config health-check)` **`copy`** *`source-alias target-alias`*
Copy from one health check to another (creating if necessary).

`#(config health-check)` **`create`** {**`composite`** *`alias_name`* | **`http`** *`alias_name url`* | **`https`** *`alias_name url`* | **`icmp`** *`alias_name hostname`* | **`ssl`** *`alias_name hostname`* [*`port`*]| **`tcp`** *`alias_name hostname`* [*`port`*]}
Create a user-defined health check of the type specified. *`Hostname`* can be an IPv4 or IPv6 host or address.

`#(config health-check)` **`default e-mail`** {**`healthy`** {**`enable`** | **`disable`**} | **`report-all-ips`** {**`enable`** | **`disable`**} | **`sick`** {**`enable`** | **`disable`**}}
Configure defaults for e-mail options.

`#(config health-check)` **`default event-log`** {**`healthy`** {**`disable`** | **`information`** | **`severe`**}| **`report-all-ips`** {**`enable`** | **`disable`**} | **`sick`**{**`information`**|**`disable`** | **`severe`**}}
Configure defaults for event-log options. An informational or a severe event-log message is logged depending on the setting chosen.

`#(config health-check)` **`default failure-trigger`** {**`none`** | *`count`*}
Configure defaults for the failure-trigger options.

`#(config health-check)` **`default interval`** {**`healthy`** *`seconds`* | **`sick`** *`seconds`*}
Configure defaults for interval options.

`#(config health-check)` **`default snmp`** {**`healthy`** {**`enable`** | **`disable`**} | **`report-all-ips`** {**`enable`** | **`disable`**} | **`sick`** {**`enable`** | **`disable`**}}
Configure defaults for snmp options.

`#(config health-check)` **`default severity`** {**`critical`** | **`no-effect`** | **`warning`**}
Configure default severity for health checks.

`#(config health-check)` **`default threshold`** {**`healthy`** *`count`* | **`response-time`** *`milliseconds`* | **`sick`** *`count`*}
Configure defaults for threshold options.

`#(config health-check)` **`delete`** *`alias_name`*
Delete the specified health check.

`#(config health-check)` **`disable`** {**`healthy`** *`alias_name`* | **`sick`** *`alias_name`*}
Disable the specified health check and have it always report health or sick.

`#(config health-check)`**`edit`** *`auth.test_name`*
Allows you to configure options for the authentication health check you specify.

`#(config health-check` *`auth.test_name`*`)` **`clear-statistics`**
Clears statistics for this health check.

`#(config health-check` *auth.test_name*`)` **e-mail** {**healthy** {**default** | **enable**| **disable**}| **report-all-ips** {**default** | **enable** | **disable**}| **sick** {**default** | **enable** | **disable**}}
Sends e-mail notification when the health check reports healthy or sick, whether or not those reports are for all IP addresses.

`#(config health-check` *auth.test_name*`)` **event-log** {**healthy** {**default** | **disable** | **information** | **severe**}| **report-all-ips** {**default** | **enable** | **disable**}| **sick** {**default** | **disable** | **information** | **severe**}}
Logs an event when the health check reports healthy or sick, whether or not those reports are for all IP addresses. An informational or a severe event-log message is logged depending on the setting chosen.

`#(config health-check` *auth.test_name*`)` **exit**
Exits the health check editing mode.

`#(config health-check` *auth.test_name*`)` **failure-trigger** {**default** | **none** | *count*}
Configures options for the failure-trigger.

`#(config health-check` *auth.test_name*`)` **interval** {**healthy** {**default** | *seconds*} | **sick** {**default** | *seconds*}}
Configures intervals before the health check is re-run. The intervals can be different for health checks that are reporting healthy and health checks that are reporting sick.

`#(config health-check` *auth.test_name*`)` **perform-health-check**
Starts the health check immediately and reports the result.

`#(config health-check` *auth.test_name)* **severity** {**critical** | **no-effect** | **default** | **warning**}
Configures default severity for the health check.

`#(config health-check` *auth.test_name*`)` **snmp** {**healthy** {**default** | **enable** | **disable**} | **report-all-ips** {**default** | **enable** | **disable**} | **sick** {**default** | **enable** | **disable**}}
Sends a trap when the health check reports healthy, whenever an IP address health check reports healthy, or when a health check reports sick.

`#(config health-check` *auth.host_name*`)` **threshold** {**healthy** {**default** | *count*} | **response-time** {**default** | **none** | *milliseconds*} | **sick** {**default** | *count*}}
Sets the level when health checks will report healthy or sick.

`#(config health-check` *auth.test_name*`)` **use-defaults**
Resets the defaults of the health check to use the global defaults instead of any explicitly set values.

`#(config health-check` *auth.test_name*`)` **view** {**configuration** | **events** | **statistics**}
Displays the health check's configuration, recent event-log messages or statistics.

`#(config health-check)` **edit** *composite_health_check*
Edit the specified composite health check.

`#(config health-check user.`*composite_health_check*`)` **add** *member_name*
Add the specified member to the composite health check group.

`#(config health-check user.`*composite_health_check*`)` **combine** {**all-healthy** | **any-healthy** | **some-healthy**}
Require that all, some, or any members of the group report as healthy to have the composite health check report as healthy.

`#(config health-check user.`*composite_health_check*`)` **e-mail** {**healthy** {**default** | **enable** | **disable**} | **report-all-ips** {**healthy** {**default** | **enable** | **disable**} | **sick** {**default** | **enable** | **disable**}}
Send e-mail notification when a health check reports healthy or sick, whether or not those reports are for all IP addresses.

`#(config health-check user.`*composite_health_check*`)` **event-log** {**healthy** {**default** | **disable** | **information** | **severe**} | **report-all-ips** {**healthy** {**default** | **enable** | **disable**} | **sick** {**default** | **enable** | **disable**}}
Log an event when a health check reports healthy or sick, whether or not those reports are for all IP addresses.

`#(config health-check user.`*composite_health_check*`)` **exit**
Leaves the composite health check editing submode.

`#(config health-check user.`*composite_health_check*`)` **perform-health-check**
Does a health check on the members of the composite immediately and reports the result.

`#(config health-check user.`*composite_health_check*`)` **remove** *member_name*
Remove a member from the composite group.

`#(config health-check user.`*composite_health_check*`)` **snmp** {**healthy** {**default** | **enable** | **disable**} | **report-all-ips** {**healthy** {**default** | **enable** | **disable**} | **sick** {**default** | **enable** | **disable**}}
Sends a trap when the health check reports healthy or sick, whether or not those reports are for all IP addresses.

`#(config health-check user.`*composite_health_check*`)` **severity** {**critical** | **default** | **no-effect** | **warning**}
Sets the severity level of the health check, which determines how this health check affects the overall health of the device.

`#(config health-check user.`*composite_health_check*`)` **use-defaults**
Re-sets the defaults of the health check to use the global defaults instead of any explicitly set values.

`#(config health-check user.`*composite_health_check*`)` **view** {**configuration** | **events** | **statistics**}
Views the composite health check's configuration, event log messages, or statistics.

`#(config health-check)` **edit** dns.*test_name*
Allows you to configure options for the DNS health check you specified.

`#(config health-check dns.`*test_name*`)` **clear-statistics**
Clears statistics for this health check.

`#(config health-check dns.`*test_name*`)` **e-mail** {**healthy** {**default** | **enable** | **disable**} | **report-all-ips** {**default** | **enable** | **disable**} | **sick** {**default** | **enable** | **disable**}}
Sends e-mail notification when the health check reports healthy or sick, whether or not those reports are for all IP addresses.

`#(config health-check dns.`*test_name*`)` **event-log** {**healthy** {**default** | **disable** | **information** | **severe**} | **report-all-ips** {**default** | **enable** | **disable**} | **sick** {**default** | **disable** | **information** | **severe**}}
Logs an event when the health check reports healthy or sick, whether or not those reports are for all IP addresses. An informational or a severe event-log message is logged depending on the setting chosen.

`#(config health-check dns.`*test_name*`)` **exit**
Exits the health check editing mode.

`#(config health-check dns.`*test_name*`)` **failure-trigger** {**default** | **none** | *count*}
Configures options for the failure-trigger.

`#(config health-check dns.`*test_name*`)` **interval** {**healthy** {**default** | *seconds*} | **sick** {**default** | *seconds*}}
Configures intervals before the health check is re-run. The intervals can be different for health checks that are reporting healthy and health checks that are reporting sick.

`#(config health-check dns.`*test_name*`)` **hostname** {**default** | *hostname*}
Sets the hostname for the DNS Server health check to the default hostname or to a user-defined hostname.

`#(config health-check dns.`*test_name*`)` **perform-health-check**
Starts the health check immediately and reports the result.

`#(config health-check dns.`*test_name*`)` **severity** {**critical** | **no-effect** | **default** | **warning**}
Configures default severity for the health check.

`#(config health-check dns.`*test_name*`)` **snmp** {**healthy** {**default** | **enable** | **disable**} | **report-all-ips** {**default** | **enable** | **disable**} | **sick** {**default** | **enable** | **disable**}}
Sends a trap when the health check reports healthy, whenever an IP address health check reports healthy, or when a health check reports sick.

`#(config health-check dns.`*test_name*`)` **threshold** {**healthy** {**default** | *count*} | **response-time** {**default** | **none** | *milliseconds*} | **sick** {**default** | *count*}}
Sets the level when health checks will report healthy or sick.

`#(config health-check dns.`*test_name*`)` **use-defaults**
Resets the defaults of the health check to use the global defaults instead of any explicitly set values.

`#(config health-check dns.`*test_name*`)` **view** {**configuration** | **events** | **statistics**}
Displays the health check's configuration, recent event-log messages or statistics.

`#(config health-check)` **edit** `drtr.`*test_name*
Allows you to configure options for the health check you specified.

`#(config health-check drtr.`*test_name*`)` **clear-statistics**
Clears statistics for this health check.

`#(config health-check drtr.`*test_name*`)` **e-mail** {**healthy** {**default** | **enable** | **disable**} | **report-all-ips** {**healthy** {**default** | **enable** | **disable**} | **sick** {**default** | **enable** | **disable**}}
Send e-mail notification when the health check reports healthy or sick, whether or not those reports are for all IP addresses.

`#(config health-check drtr.`*test_name*`)` **event-log** {**healthy** {**default** | **disable** | **information** | **severe**} | **report-all-ips** {**healthy** {**default** | **enable** | **disable**} | **sick** {**default** | **enable** | **disable**}}
Log an event when the health check reports healthy or sick, whether or not those reports are for all IP addresses.

`#(config health-check drtr.`*test_name*`)` **exit**
Leaves the health check editing mode.

`#(config health-check drtr.`*test_name*`)` **failure-trigger** {**default** | **none** | *count*}
Configure options for the failure-trigger.

`#(config health-check drtr.`*test_name*`)` **interval** {**healthy** {**default** | *seconds*} | **sick** {**default** | *seconds*}}
Configure intervals before the health check is re-run. The intervals can be different for health checks that are reporting healthy and health checks that are reporting sick.

`#(config health-check drtr.`*test_name*`)` **perform-health-check**
Starts the health check immediately and reports the result.

`#(config health-check drtr.`*test_name*`)` **snmp** {**healthy** {**default** | **enable** | **disable**} | **report-all-ips** {**healthy** {**default** | **enable** | **disable**} | **sick** {**default** | **enable** | **disable**}
Sends a trap when the health check reports healthy, whenever an IP address health check reports healthy, or when a health check reports sick.

`#(config health-check drtr.`*`test_name`*`)` **`threshold`** {**`healthy`** {**`default`** | *`count`*} | **`response-time`** {**`default`** | **`none`** | *`milliseconds`*} | **`sick`** {**`default`** | *`count`*}}
Set the level when health checks will report healthy or sick.

`#(config health-check drtr.`*`test_name`*`)` **`use-defaults`**
Re-sets the defaults of the health check to use the global defaults instead of any explicitly set values.

`#(config health-check drtr.`*`test_name`*`)` **`view`** {**`configuration`** | **`statistics`**}
Views the health check's configuration or statistics.

`#(config health-check)` **`edit`** `fwd.`*`group_name`*
Allows you to configure options for the health check you specified.

`#(config health-check fwd.`*`group_name`*`)` **`combine`** {**`all healthy`** | **`any-healthy`** | **`some-healthy`**}
Combines the results when a group test is healthy.

`#(config health-check fwd.`*`group_name`*`)` **`e-mail`** {**`healthy`** {**`default`** | **`enable`** | **`disable`**} | **`report-all-ips`** {**`healthy`** {**`default`** | **`enable`** | **`disable`**} | **`sick`** {**`default`** | **`enable`** | **`disable`**}}
Send e-mail notification when the health check reports healthy or sick, whether or not those reports are for all IP addresses.

`#(config health-check fwd.`*`group_name`*`)` **`event-log`** {**`healthy`** {**`default`** | **`disable`** | **`information`** | **`severe`**} | **`report-all-ips`** {**`healthy`** {**`default`** | **`enable`** | **`disable`**} | **`sick`** {**`default`** | **`enable`** | **`disable`**}}
Log an event when the health check reports healthy or sick, whether or not those reports are for all IP addresses.

`#(config health-check fwd.`*`group_name`*`)` **`exit`**
Leaves the health check editing mode.

`#(config health-check fwd.`*`group_name`*`)` **`perform-health-check`**
Starts the health check immediately and reports the result.

`#(config health-check fwd.`*`group_name`*`)` **`snmp`** {**`healthy`** {**`default`** | **`enable`** | **`disable`**} | **`report-all-ips`** {**`healthy`** {**`default`** | **`enable`** | **`disable`**} | **`sick`** {**`default`** | **`enable`** | **`disable`**}}
Sends a trap when the health check reports healthy, whenever an IP address health check reports healthy, or when a health check reports sick.

`#(config health-check fwd.`*`group_name`*`)` **`use-defaults`**
Re-sets the defaults of the health check to use the global defaults instead of any explicitly set values.

`#(config health-check fwd.`*`group_name`*`)` **`view`** {**`configuration`** | **`statistics`**}
Views the health check's configuration or statistics.

`#(config health-check)` **`edit`** `fwd.`*`host_name`*
Allows you to configure options for the health check you specified.

`#(config health-check fwd.`*`host_name`*`)` **`authentication`** {**`basic`** | **`disable`** | **`encrypted-password`** *`encrypted-password`* | **`password`** *`password`* | **`username`** *`username`*}
(Used with HTTP or HTTPS health checks.) To test Basic authentication, you can enter the username and password of the target.

`#(config health-check fwd.`*`host_name`*`)` **`clear-statistics`**
Clears statistics for this health check.

`#(config health-check fwd.`*`host_name`*`)` **`e-mail`** {**`healthy`** {**`default`** | **`enable`** | **`disable`**} | **`report-all-ips`** {**`healthy`** {**`default`** | **`enable`** | **`disable`**} | **`sick`** {**`default`** | **`enable`** | **`disable`**}}
Send e-mail notification when the health check reports healthy or sick, whether or not those reports are for all IP addresses.

`#(config health-check fwd.`*`host_name`*`)` **`event-log`** {**`healthy`** {**`default`** | **`disable`** | **`information`** | **`severe`**} | **`report-all-ips`** {**`healthy`** {**`default`** | **`enable`** | **`disable`**} |

sick {**default** | **enable** | **disable**}}
Log an event when the health check reports healthy or sick, whether or not those reports are for all IP addresses.

`#(config health-check fwd.`*host_name*`)` **exit**
Leaves the health check editing mode.

`#(config health-check fwd.`*host_name*`)` **failure-trigger** {**default** | **none** | *count*}
Configure options for the failure-trigger.

`#(config health-check fwd.`*host_name*`)` **interval** {**healthy** {**default** | *seconds*}| **sick** {**default** | *seconds*}}
Configure intervals before the health check is re-run. The intervals can be different for health checks that are reporting healthy and health checks that are reporting sick.

`#(config health-check fwd.`*host_name*`)` **perform-health-check**
Starts the health check immediately and reports the result.

`#(config health-check fwd.`*host_name*`)` **proxy-authentication** {**basic** | **disable** | **encrypted-password** *encrypted-password* | **password** *password* | **username** *username*}
(Used with HTTP or HTTPS health checks, when intermediate proxies are between you and the target.) Enter the username and password of the intermediate proxy.

`#(config health-check fwd.`*host_name*`)` **response-code** {**add** *codes* | **remove** *codes*}
To manage a list of codes that are considered successes, you can add or remove codes, separated by semi-colons. If a success code is received by the health check, the health check considers the HTTP/HTTPS test to be successful.

`#(config health-check fwd.`*host_name*`)` **snmp** {**healthy** {**default** | **enable** | **disable**}| **report-all-ips** {**healthy** {**default** | **enable** | **disable**}| **sick** {**default** | **enable** | **disable**}}
Sends a trap when the health check reports healthy, whenever an IP address health check reports healthy, or when a health check reports sick.

`#(config health-check fwd.`*host_name*`)` **threshold** {**healthy** {**default** | *count*} | **response-time** {**default** | **none** | *milliseconds*} | **sick** {**default** | *count*}}
Set the level when health checks will report healthy or sick.

`#(config health-check fwd.`*host_name*`)` **type** (**http** *URL* | **https** *URL* | **icmp** *hostname* | **ssl** *hostname* [*port*] | **tcp** *hostname* [*port*]}
Set the number of consecutive healthy or sick test results before the health check actually reports as healthy or sick.

`#(config health-check fwd.`*host_name*`)` **use-defaults**
Re-sets the defaults of the health check to use the global defaults instead of any explicitly set values.

`#(config health-check fwd.`*host_name*`)` **view** {**configuration** | **statistics**}
Views the health check's configuration or statistics.

`#(config health-check)` **edit** *health_check_name*
Allows you to configure options for the health check you specified.

`#(config health-check user.`*health_check_name*`)` **authentication** {**basic** | **disable** | **encrypted-password** *encrypted-password*| **password** *password*| **username** *username*}
(Used with HTTP or HTTPS health checks.) To test Basic authentication, you can enter the username and password of the target.

`#(config health-check user.`*health_check_name*`)` **clear-statistics**
Clears statistics for this health check.

`#(config health-check user.`*health_check_name*`)` **e-mail** {**healthy** {**default** | **enable** | **disable**}| **report-all-ips** {**healthy** {**default** | **enable** | **disable**}| **sick** {**default** | **enable** | **disable**}}

Send e-mail notification when the health check reports healthy or sick, whether or not those reports are for all IP addresses.

`#(config health-check user.`*health_check_name*`)` **event-log** {**healthy** {**default** | **disable** |**information** | **severe**}| **report-all-ips** {**healthy** {**default** | **enable** | **disable**}| **sick** {**default** | **enable** | **disable**}}
Log an event when the health check reports healthy or sick, whether or not those reports are for all IP addresses.

`#(config health-check user.`*health_check_name*`)` **exit**
Leaves the health check editing mode.

`#(config health-check user.`*health_check_name*`)` **failure-trigger** {**default** | **none** | *count*}
Configure options for the failure-trigger.

`#(config health-check user.`*health_check_name*`)` **interval** {**healthy** {**default** | *seconds*}| **sick** {**default** | *seconds*}}
Configure intervals before the health check is re-run. The intervals can be different for health checks that are reporting healthy and health checks that are reporting sick.

`#(config health-check user.`*health_check_name*`)` **perform-health-check**
Starts the health check immediately and reports the result.

`#(config health-check user.`*health_check_name*`)` **proxy-authentication** {**basic** | **disable** | **encrypted-password** *encrypted-password* | **password** *password* | **username** *username*}
(Used with HTTP or HTTPS health checks, when intermediate proxies are between you and the target.) Enter the username and password of the intermediate proxy.

`#(config health-check user.`*health_check_name*`)` **response-code** {**add** *codes* | **remove** *codes*}
To manage a list of codes that are considered successes, you can add or remove codes, separated by semi-colons. If a success code is received by the health check, the health check considers the HTTP/HTTPS test to be successful.

`#(config health-check user.`*health_check_name*`)` **snmp** {**healthy** {**default** | **enable** | **disable**}| **report-all-ips** {**healthy** {**default** | **enable** | **disable**}| **sick** {**default** | **enable** | **disable**}}
Sends a trap when the health check reports healthy, whenever an IP address health check reports healthy, or when a health check reports sick.

`#(config health-check user.`*health_check_name*`)` **threshold** {**healthy** {**default** | *count*} | **response-time** {**default** | **none** | *milliseconds*} | **sick** {**default** | *count*}}
Set the level when health checks will report healthy or sick.

`#(config health-check user.`*health_check_name*`)` **type** (**http** *URL* | **https** *URL* | **icmp** *hostname* | **ssl** *hostname* [*port*] | **tcp** *hostname* [*port*]}
Set the number of consecutive healthy or sick test results before the health check actually reports as healthy or sick.

`#(config health-check user.`*health_check_name*`)` **use-defaults**
Re-sets the defaults of the health check to use the global defaults instead of any explicitly set values.

`#(config health-check user.`*health_check_name*`)` **view** {**configuration** | **statistics**}
Views the health check's configuration or statistics.

`#(config health-check)` **edit** `icap.`*test_name*
Allows you to configure options for the health check you specified.

`#(config health-check icap.`*test_name*`)` **clear-statistics**
Clears statistics for this health check.

`#(config health-check icap.`*test_name*`)` **e-mail** {**healthy** {**default** | **enable** | **disable**}| **report-all-ips** {**healthy** {**default** | **enable** | **disable**}| **sick** {**default** |

enable | **disable**}}
Send e-mail notification when the health check reports healthy or sick, whether or not those reports are for all IP addresses.

#(config health-check icap.*test_name*) **event-log** {**healthy** {**default** | **disable** |**information** | **severe**}| **report-all-ips** {**healthy** {**default** | **enable** | **disable**}| **sick** {**default** | **enable** | **disable**}}
Log an event when the health check reports healthy or sick, whether or not those reports are for all IP addresses.

#(config health-check icap.*test_name*) **exit**
Leaves the health check editing mode.

#(config health-check icap.*test_name*) **failure-trigger** {**default** | **none** | *count*}
Configure options for the failure-trigger.

#(config health-check icap.*test_name*) **interval** {**healthy** {**default** | *seconds*}| **sick** {**default** | *seconds*}}
Configure intervals before the health check is re-run. The intervals can be different for health checks that are reporting healthy and health checks that are reporting sick.

#(config health-check icap.*test_name*) **perform-health-check**
Starts the health check immediately and reports the result.

#(config health-check icap.*test_name*) **snmp** {**healthy** {**default** | **enable** | **disable**}| **report-all-ips** {**healthy** {**default** | **enable** | **disable**}| **sick** {**default** | **enable** | **disable**}}
Sends a trap when the health check reports healthy, whenever an IP address health check reports healthy, or when a health check reports sick.

#(config health-check icap.*test_name*) **threshold** {**healthy** {**default** | *count*} | **response-time** {**default** | **none** | *milliseconds*} | **sick** {**default** | *count*}}
Set the level when health checks will report healthy or sick.

#(config health-check icap.*test_name*) **use-defaults**
Re-sets the defaults of the health check to use the global defaults instead of any explicitly set values.

#(config health-check icap.*test_name*) **view** {**configuration** | **statistics**}
Views the health check's configuration or statistics.

#(config health-check) **edit** socks.*test_name*
Allows you to configure options for the health check you specified.

#(config health-check socks.*test_name*) **clear-statistics**
Clears statistics for this health check.

#(config health-check socks.*test_name*) **e-mail** {**healthy** {**default** | **enable** | **disable**}| **report-all-ips** {**healthy** {**default** | **enable** | **disable**}| **sick** {**default** | **enable** | **disable**}}
Send e-mail notification when the health check reports healthy or sick, whether or not those reports are for all IP addresses.

#(config health-check socks.*test_name*) **event-log** {**healthy** {**default** | **disable** |**information** | **severe**}| **report-all-ips** {**healthy** {**default** | **enable** | **disable**}| **sick** {**default** | **enable** | **disable**}}
Log an event when the health check reports healthy or sick, whether or not those reports are for all IP addresses.

#(config health-check socks.*test_name*) **exit**
Leaves the health check editing mode.

#(config health-check socks.*test_name*) **failure-trigger** {**default** | **none** | *count*}
Configure options for the failure-trigger.

`#(config health-check socks.test_name)` **interval** {**healthy** {**default** | *seconds*}| **sick** {**default** | *seconds*}}
Configure intervals before the health check is re-run. The intervals can be different for health checks that are reporting healthy and health checks that are reporting sick.

`#(config health-check socks.test_name)` **perform-health-check**
Starts the health check immediately and reports the result.

`#(config health-check socks.test_name)` **snmp** {**healthy** {**default** | **enable** | **disable**}| **report-all-ips** {**healthy** {**default** | **enable** | **disable**}| **sick** {**default** | **enable** | **disable**}}
Sends a trap when the health check reports healthy, whenever an IP address health check reports healthy, or when a health check reports sick.

`#(config health-check socks.test_name)` **threshold** {**healthy** {**default** | *count*} | **response-time** {**default** | **none** | *milliseconds*} | **sick** {**default** | *count*}}
Set the level when health checks will report healthy or sick.

`#(config health-check socks.test_name)` **type** (**http** *URL* | **https** *URL* | **icmp** *hostname* | **ssl** *hostname* [*port*] | **tcp** *hostname* [*port*]}
Set the number of consecutive healthy or sick test results before the health check actually reports as healthy or sick.

`#(config health-check socks.test_name)` **use-defaults**
Re-sets the defaults of the health check to use the global defaults instead of any explicitly set values.

`#(config health-check socks.test_name)` **view** {**configuration** | **statistics**}
Views the health check's configuration or statistics.

`#(config health-check)` **edit** `ws.test_name`
Allows you to configure options for the health check you specified.

`#(config health-check ws.test_name)` **clear-statistics**
Clears statistics for this health check.

`#(config health-check ws.test_name)` **e-mail** {**healthy** {**default** | **enable** | **disable**}| **report-all-ips** {**healthy** {**default** | **enable** | **disable**}| **sick** {**default** | **enable** | **disable**}}
Send e-mail notification when the health check reports healthy or sick, whether or not those reports are for all IP addresses.

`#(config health-check ws.test_name)` **event-log** {**healthy** {**default** | **disable** |**information** | **severe**}| **report-all-ips** {**healthy** {**default** | **enable** | **disable**}| **sick** {**default** | **enable** | **disable**}}
Log an event when the health check reports healthy or sick, whether or not those reports are for all IP addresses.

`#(config health-check ws.test_name)` **exit**
Leaves the health check editing mode.

`#(config health-check ws.test_name)` **failure-trigger** {**default** | **none** | *count*}
Configure options for the failure-trigger.

`#(config health-check ws.test_name)` **interval** {**healthy** {**default** | *seconds*}| **sick** {**default** | *seconds*}}
Configure intervals before the health check is re-run. The intervals can be different for health checks that are reporting healthy and health checks that are reporting sick.

`#(config health-check ws.test_name)` **perform-health-check**
Starts the health check immediately and reports the result.

`#(config health-check ws.`*test_name*`)` **snmp** {**healthy** {**default** | **enable** | **disable**}| **report-all-ips** {**healthy** {**default** | **enable** | **disable**}| **sick** {**default** | **enable** | **disable**}}
Sends a trap when the health check reports healthy, whenever an IP address health check reports healthy, or when a health check reports sick.

`#(config health-check ws.`*test_name*`)` **test-url** {**default** | *url*}
Sets the test URL to default.

`#(config health-check ws.`*test_name*`)` **threshold** {**healthy** {**default** | *count*} | **response-time** {**default** | **none** | *milliseconds*} | **sick** {**default** | *count*}}
Set the level when health checks will report healthy or sick.

`#(config health-check ws.`*test_name*`)` **use-defaults**
Re-sets the defaults of the health check to use the global defaults instead of any explicitly set values.

`#(config health-check ws.`*test_name*`)` **view** {**configuration** | **statistics**}
Views the health check's configuration or statistics.

`#(config health-check)` **edit** `ws.`*group_name*
Allows you to configure options for the health check you specified.

`#(config health-check ws.`*group_name*`)` **combine** {**all healthy** | **any-healthy** | **some-healthy**}
Combines the results when a group test is healthy.

`#(config health-check ws.`*group_name*`)` **e-mail** {**healthy** {**default** | **enable** | **disable**}| **report-all-ips** {**healthy** {**default** | **enable** | **disable**}| **sick** {**default** | **enable** | **disable**}}
Send e-mail notification when the health check reports healthy or sick, whether or not those reports are for all IP addresses.

`#(config health-check ws.`*group_name*`)` **event-log** {**healthy** {**default** | **disable** |**information** | **severe**}| **report-all-ips** {**healthy** {**default** | **enable** | **disable**}| **sick** {**default** | **enable** | **disable**}}
Log an event when the health check reports healthy or sick, whether or not those reports are for all IP addresses.

`#(config health-check ws.`*group_name*`)` **exit**
Leaves the health check editing mode.

`#(config health-check ws.`*group_name*`)` **perform-health-check**
Starts the health check immediately and reports the result.

`#(config health-check ws.`*group_name*`)` **snmp** {**healthy** {**default** | **enable** | **disable**}| **report-all-ips** {**healthy** {**default** | **enable** | **disable**}| **sick** {**default** | **enable** | **disable**}}
Sends a trap when the health check reports healthy, whenever an IP address health check reports healthy, or when a health check reports sick.

`#(config health-check ws.`*group_name*`)` **use-defaults**
Re-sets the defaults of the health check to use the global defaults instead of any explicitly set values.

`#(config health-check ws.`*group_name*`)` **view** {**configuration** | **statistics**}
Views the health check's configuration or statistics.

`#(config health-check)` **enable** *alias_name*
Enable the health check of the specified name.

`#(config health-check)` **exit**
Leave the health-check configuration mode.

`#(config health-check)` **perform-health-check** *alias_name*
Runs the specified health check.

`#(config health-check)` **`view {configuration | quick-statistics | statistics}`**
Views the configuration or statistics for all health checks. You can also view a summary of the health-check statistics.

For More Information

- *SGOS Administration Guide*

Example

```
SGOS#(config) health-check
SGOS#(config health-check) create composite composite1
SGOS#(config health-check) edit composite1
SGOS#(config health-check user.composite1) view statistics
Enabled     Health check failed     DOWN
```

#(config) hide-advanced

See

- **# hide-advanced** on page 57.

#(config) http

Synopsis

Use this command to configure HTTP settings.

Syntax

`#(config)` **`http [no] add-header client-ip`**
Adds the `client-ip` header to forwarded requests.

`#(config)` **`http [no] add-header front-end-https`**
Adds the `front-end-https` header to forwarded requests.

`#(config)` **`http [no] add-header via`**
Adds the `via` header to forwarded requests.

`#(config)` **`http [no] add-header x-forwarded-for`**
Adds the `x-forwarded-for` header to forwarded requests.

`#(config)` **`http [no] byte-ranges`**
Enables HTTP byte-range support.

If byte-range support is disabled, then HTTP treats all byte range requests as non-cacheable. This means that HTTP never even checks to see if the object is in the cache, but forwards the request to the origin-server and does not cache the result. So the range request has no affect on the cache. For instance, if the object was in the cache before a range request, it would still be in the cache afterward—the range request does not delete any currently cached objects. Also, the Range header is not modified when forwarded to the origin-server.

If the requested byte range is type 3 or 4, then the request is treated as if byte-range support is disabled. That is, the request is treated as non-cacheable and has no affect on objects in the cache.

`#(config)` **`http [no] cache authenticated-data`**
Caches any data that appears to be authenticated.

`#(config)` **`http [no] cache expired`**
Retains cached objects older than the explicit expiration.

`#(config)` **`http [no] cache personal-pages`**
Caches objects that appear to be personal pages.

`#(config)` **`http [no] clientless-requests`**
Limits the number of clientless requests (used for caching and optimization) and prevent overwhelming an OCS.

`#(config)` **`http [no] exception-on-network-error`**
Using the no option prevents the ProxySG from sending exception pages to clients when upstream connection errors occur.

`#(config)` **`http [no] force-ntlm`**
Uses NTLM for Microsoft Internet Explorer proxy.

`#(config)` **`http ftp-proxy-url root-dir`**
URL path is absolute in relation to the root.

`#(config)` **`http ftp-proxy-url user-dir`**
URL path is relative to the user's home directory.

`#(config)` **`http [no] location-header-rewrite`**
Auto rewrite location header in reverse proxy.

`#(config)` **`http [no] parse meta-tag {cache-control | expires | pragma-no-cache}`**
Parses HTML objects for the `cache-control`, `expires`, and `pragma-no-cache` meta-tags.

`#(config)` **`http [no] persistent client`**
Enables support for persistent client requests from the browser.

`#(config)` **`http [no] persistent server`**
Enables support for persistent server requests to the Web server.

`#(config)` **`http [no] persistent-timeout client`** *`num_seconds`*
Sets persistent connection timeout for the client to *`num_seconds`*.

`#(config)` **`http [no] persistent-timeout server`** *`num_seconds`*
Sets persistent connection timeout for the server to *`num_seconds`*.

`#(config)` **`http [no] pipeline client {requests | redirects}`**
Prefetches either embedded objects in client requests or redirected responses to client requests.

`#(config)` **`http [no] pipeline prefetch {requests | redirects}`**
Prefetches either embedded objects in pipelined objects or redirected responses to pipelined requests.

`#(config)` **`http [no] proprietary-headers bluecoat`**
`Enables the Blue Coat proprietary HTTP header extensions.`

`#(config)` **`http receive-timeout client`** *`num_seconds`*
Sets receive timeout for client to *`num_seconds`*.

`#(config)` **`http receive-timeout refresh`** *`num_seconds`*
Sets receive timeout for refresh to *`num_seconds`*.

`#(config)` **`http receive-timeout server`** *`num_seconds`*
Sets receive timeout for server to *`num_seconds`*.

`#(config)` **`http [no] revalidate-pragma-no-cache`**
Revalidates "Pragma: no-cache."

`#(config)` **`http [no] strict-expiration refresh`**
Forces compliance with explicit expirations by never refreshing objects before their explicit expiration.

`#(config)` **`http [no] strict-expiration serve`**
Forces compliance with explicit expirations by never serving objects after their explicit expiration.

`#(config)` **`http [no] strip-from-header`**
Removes HTTP information from headers.

`#(config)` **`http [no] substitute conditional`**
Uses an HTTP "get" in place of HTTP 1.1 conditional get.

`#(config)` **`http [no] substitute ie-reload`**
Uses an HTTP "get" for Microsoft Internet Explorer reload requests.

`#(config)` **`http [no] substitute if-modified-since`**
Uses an HTTP "get" instead of "get-if-modified."

`#(config)` **`http [no] substitute pragma-no-cache`**
Uses an HTTP "get" instead of "get pragma: no-cache."

`#(config)` **`http [no] tolerant-request-parsing`**
Enables or disables the HTTP tolerant-request-parsing flag.

`#(config)` **`http upload-with-pasv disable`**
Disables uploading with Passive FTP.

`#(config)` **`http upload-with-pasv enable`**
Enables uploading with Passive FTP.

`#(config)` **`http version {1.0 | 1.1 | preserve}`**
Indicates the version of HTTP that should be used by the ProxySG. The `preserve` option preserves the inbound HTTP version.

`#(config)` **`http [no] www-redirect`**
Redirects to `www.`*`host`*`.com` if host not found.

`#(config)` **`http [no] xp-rewrite-redirect`**
Rewrites origin server 302s to 307s for Windows XP IE requests.

For More Information

- **`#(config http-console)`** on page 240
- **`#(config HTTP)`** on page 288
- *SGOS Administration Guide*

#(config) identd

Synopsis

IDENTD implements the TCP/IP IDENT user identification protocol. IDENTD operates by looking up specific TCP/IP connections and returning the user name of the process owning the connection.

Syntax

```
#(config) identd
```

This changes the prompt to:

```
#(config identd)
```

Subcommands

`#(config identd) client server-query-port` *port*
Specifies the port to query on the client machines. The default is 113.

`#(config identd) client timeout` *seconds*
Specifies the timeout in seconds for identd queries. The default is 30 seconds.

`#(config identd) trim-whitespace {enable | disable}`
Specify whether to trim leading and trailing whitespace in the username portion of the identd query response. By default this is disabled.

If client identd servers are adding insignificant whitespace to the username field you might need to enable this option to trim the username as expected.

`#(config identd) exit`
Exits configure identd mode and returns to configure mode.

`#(config identd) server {enable | disable}`
Enables or disables identd services.

`#(config identd) view`
Displays current identd settings.

For More Information

- *SGOS Administration Guide*

Example

```
SGOS#(config) identd
SGOS#(config identd) enable
 ok
SGOS#(config identd) exit
SGOS#(config)
```

#(config) inline

See

- **# inline** on page 58

#(config) installed-systems

Synopsis

Use this command to manage the list of installed ProxySG systems.

Syntax

```
#(config) installed-systems
```

This changes the prompt to:

```
#(config installed-systems)
```

Subcommands

`#(config installed-systems)` **`default`** *`system_number`* **`[ignore-warnings]`**
Sets the default system to the system indicated by *`system_number`*. The `ignore-warnings` option allows you to set the default system even if you receive a disk layout compatibility warning. Keep in mind that if you use the `ignore-warnings` option to forcing a change to a default system that is incompatible with your disk layout may result in configuration and/or data loss.

`#(config installed-systems)` **`delete`** *`system_number`*
Deletes the system indicated by *`system_number`*.

`#(config installed-systems)` **`enforce-signed`** {**`enable`** | **`disable`**}
Restricts system image download and installation to signed images only. The default, `disable`, allows all images to be downloaded.

`#(config installed-systems)` **`exit`**
Exits configure installed-systems mode and returns to configure mode.

`#(config installed-systems)` **`lock`** *`system_number`*
Locks the system indicated by *`system_number`*.

`#(config installed-systems)` **`no`** {**`lock`** *`system_number`* | **`replace`**}
lock *`system_number:`* Unlocks the system indicated by *`system_number`* if it is currently locked. replace: Specifies that the system currently tagged for replacement should not be replaced. The default replacement is used (oldest unlocked system).

`#(config installed-systems)` **`replace`** *`system_number`*
Specifies that the system identified by *`system_number`* is to be replaced next.

`#(config installed-systems)` **`view`**
Shows installed ProxySG systems.

For More Information

- *SGOS Administration Guide*

Example

```
SGOS#(config) installed-systems
SGOS#(config installed-systems) default 2
 ok
SGOS#(config installed-systems) lock 1
 ok
SGOS#(config installed-systems) exit
SGOS#(config)
```

#(config) interface

Synopsis

This command enables you to configure the network interfaces (both physical and Virtual LAN).

The built-in Ethernet adapter is configured for the first time using the setup console. If you want to modify the built-in adapter configuration, or if you have multiple adapters, you can configure each one using the command-line interface.

Syntax

`#(config)` **`interface fast-ethernet`** *`interface_number`*
where *`interface_number`* sets the number of the fast Ethernet connection to *`interface_number`*. Valid values for *`interface_number`* are 0 through 3, inclusive.

`#(config)` **`interface`** *`adapter_number`*`:`*`interface_number`*
This changes the prompt to `#(config interface 0:00`, for example

`#(config)` **`interface`** *`adapter_number`*`:`*`interface_number.vlan_id`*
Allows you to associate VLAN identification numbers with a physical interfaces.

#(config interface interface_number)

Syntax

`#(config) interface` *interface_number*

This changes the prompt to `#(config interface` *interface_number*)

Subcommands

`#(config interface` *interface_number*`) allow-intercept {enable | disable}`
Allows interception on this interface.

`#(config interface` *interface_number*`) clear-all-vlans`
Resets all VLAN parameters to their default values.

`#(config interface` *interface_number*`) exit`
Exits `#(config interface number)` mode and returns to `#(config)` mode.

`#(config interface` *interface_number*`) full-duplex`
Configures the interface for full-duplex.

`#(config interface` *interface_number*`) half-duplex`
Configures the interface for half-duplex.

`#(config interface` *interface_number*`) ip-address` *ip-address* [*subnet_mask_for IPv4*] | [*prefix_length_for_IPv6*]
Sets the IPv4 address and subnet mask or IPv6 address and prefix length for this interface.

`#(config interface` *interface_number*`) ipv6 auto-linklocal {enable | disable}`
Enables or disables the automatic generation of link-local addresses for this interface. After a link-local address is generated for an interface, it will stay configured until it is manually removed using the `no ip-address` command or until the ProxySG is rebooted.

`#(config interface` *interface_number*`) label` *label_name*
Give the interface a name for easy identification.

`#(config interface` *interface_number*`) link-autosense {enable | disable}`
Specifies that the interface should autosense speed and duplex.

`#(config interface` *interface_number*`) mtu-size` *size*
Specifies the MTU (maximum transmission unit) size.

Jumbo frames are Ethernet frames with more than 1500 bytes of payload. By configuring the mtu size of an interface to 1500 bytes or more you are enabling jumbo frames. You can configure jumbo frames between 1500 and 9000 MTUs. The max mtu size depends on the NIC you are using. If you have attempted to configure the mtu size to one that is not compatible with the NIC you are using, an error message will be displayed.

Notes

- If ProxySG receives frames over 1500 bytes, but the mtu size has not been set to enable jumbo frames, packets may be dropped.
- Configure the interfaces on a software bridge with identical MTU settings. Using different MTU interface settings on a bridge can cause unpredictable behavior.

`#(config interface` *interface_number*`) native-vlan` *number*
Sets the native VLAN value for this interface.

`#(config interface` *interface_number*`) no {ip-address | label}`
Removes the IP address or label from the interface.

`#(config interface` *`interface_number`*`)` **`reject-inbound {enable | disable}`**
Rejects inbound connections on the interface.

`#(config interface` *`interface_number`*`)` **`speed {10 | 100 | 1gb | 10gb }`**
`Specifies the interface speed.`

`#(config interface` *`interface_number`*`)` **`vlan-trunk {enable | disable}`**
Enables VLAN trunking on this interface.

`#(config interface` *`interface_number`*`)` **`view`**
Displays the interface settings.

*The `allow-intercept and reject-inbound` commands are interface-level configurations and are not bridge-specific. The `reject-inbound` command always has precedence.

The following table describes how traffic is handled for the three possible settings of these options.

reject-inbound	allow-intercept	Non-proxy ports (mgmt-console, ssh, etc)	Explicit proxy ports	Transparent proxy ports	Other ports
Disabled	Enabled	Terminated	Terminated	Terminated	Forwarded
Disabled	Disabled	Terminated	Terminated	Forwarded	Forwarded
Enabled	Enabled/Disabled	Silently dropped	Silently dropped	Silently dropped	Silently dropped

For More Information

- *SGOS Administration Guide*

Example

```
#(config) interface 0
#(config interface 0) ip-address 10.252.10.54 255.255.255.0
 ok
#(config interface 0) exit
SGOS#(config) interface 0:1
#(config interface 0:1) 10.252.10.72
ok
#(config interface 0:1) exit
```

#(config) ip-default-gateway

Synopsis

A key feature of the ProxySG is the ability to distribute traffic originating at the cache through multiple IP gateways. Further, you can fine tune how the traffic is distributed among gateways. This feature works with any routing protocol (for example, static routes or RIP).

Note: Load balancing through multiple IP gateways is independent from the per-interface load balancing that the ProxySG automatically does when more than one network interface is installed.

Syntax

`#(config) ip-default-gateway` *ip_address* `[preference group (1-10)] [weight (1-100)]`

Specifies the IPv4 or IPv6 address of the default gateway to be used by the ProxySG.

For More Information

- *SGOS Administration Guide*

Example

```
SGOS#(config) ip-default-gateway 10.25.36.47
 ok
```

#(config) ipv6

Synopsis

Use this command to configure IPv6 global settings.

Syntax

```
#(config) ipv6
```

Subcommands

`#(config)` **`ipv6 auto-linklocal {enable | disable}`**

Enable or disable automatic generation of link-local addresses on all interfaces. When this parameter is enabled (as it is by default), individual interface configuration values will override this setting. When this setting is disabled, it will be disabled for all interfaces (regardless of the per-interface setting). After link-local addresses are generated for the `ProxySG` interfaces, they will stay configured until they are manually removed using the `no ip-address` command or until the `ProxySG` is rebooted.

`#(config)` **`ipv6 force-bypass {enable | disable}`**

Enable or disable IPv6 force-bypass. When force-bypass is enabled, all IPv6 traffic will be bridged or routed. This option is disabled by default.

`#(config)` **`ipv6 forwarding {enable | disable}`**

Enable or disable IPv6 forwarding. This is a layer-3 configuration. When IPv6 forwarding is disabled (as it is by default), the `ProxySG` will discard bypassed traffic at the IPv6 layer; this setting is appropriate for most situations, since by default, the `ProxySG` is not configured to function as a router.

For More Information

- *SGOS Administration Guide, Using the* `ProxySG` *in an IPv6 Environment*

Example

```
SGOS#(config)ipv6 auto-linklocal disable
ok
```

#(config) isatap

Synopsis

Use this command to configure ProxySG behavior with Intra-Site Automatic Tunnel Addressing Protocol (ISATAP) traffic. When ISATAP is enabled with the `isatap` commands, the ProxySG looks inside the encapsulated 6-in-4 packet to determine the service and then chooses a proxy to use:

- If the service is intercepted, the ISATAP traffic is processed by the appropriate application proxy (HTTP, CIFS, FTP, etc.). Traffic is optimized with all applicable acceleration techniques and sent through an ADN tunnel if an ADN peer is found.
- If the service is not intercepted, the traffic is processed by the ISATAP proxy. Traffic is optimized with byte caching and compression inside an ADN tunnel (assuming an ADN peer is found).

Syntax

```
#(config) isatap
```

Subcommands

`#(config)` **`isatap allow-intercept {enable | disable}`**
When this command is enabled, 6-in-4 packets of intercepted services are processed by the appropriate application proxy (for example, CIFS, HTTP, or Flash). When `allow-intercept` is disabled, this traffic is processed by the ISATAP proxy. For full ISATAP functionality, enable `allow-intercept` and `adn-tunnel` (see next command).

`#(config)` **`isatap adn-tunnel {enable | disable}`**
When this command is enabled, the ISATAP proxy processes 6-in-4 traffic for services (such as ICMPv6) that aren't intercepted. When `adn-tunnel` is disabled, the ISATAP proxy is not used: any traffic that would have been processed by this proxy is bypassed. For full ISATAP functionality, enable `adn-tunnel` and `allow-intercept` (see previous command).

`#(config)` **`isatap adn-tunnel adn-byte-cache {disable | enable}`**
This command applies to traffic that the ISATAP proxy is processing. It controls whether to optimize this traffic using the byte caching optimization technique when connecting upstream in an ADN tunnel. This option is enabled by default.

`#(config)` **`isatap adn-tunnel adn-compress {disable | enable}`**
This command applies to traffic that the ISATAP proxy is processing. It controls whether to optimize this traffic using GZIP compression when connecting upstream in an ADN tunnel. This option is enabled by default.

`#(config)` **`isatap adn-tunnel byte-cache-priority {low | normal | high}`**
This command applies to traffic that the ISATAP proxy is processing. You can adjust retention priority of byte cache data. If you want to keep streams in the byte cache for as long as possible, set a high retention priority. Or for streams that aren't likely to get much benefit from byte caching, you can set a low retention priority. ISATAP is set to normal priority by default. Note that unless `adn-byte-cache` is enabled for ISATAP, the priority setting will have no effect; if you try to set a retention priority when byte caching is disabled, a warning message displays to inform you that the `byte-cache-priority` attribute has no effect when `adn-byte-cache` is disabled.

For More Information

- *SGOS Administration Guide, Using the* **ProxySG** *in an IPv6 Environment*

Example

To enable ISATAP:

```
SGOS#(config)isatap allow-intercept enable
ok
SGOS#(config)isatap adn-tunnel enable
```

#(config) license-key

Synopsis

Use this command to configure license key settings.

Syntax

`#(config)` **`license-key auto-update {disable | enable}`**
Disables or enables auto-update of the `Blue Coat` license key.

`#(config)` **`license-key no path`**
Negates certain license key settings.

`#(config)` **`license-key path`** *`url`*
Specifies the network path to download the license key.

For More Information

- *SGOS Administration Guide*

Example

```
SGOS#(config) license-key no path
 ok
```

#(config) line-vty

Synopsis

When you have a CLI session, that session remains open as long as there is activity. If you leave the session idle, the connection eventually times out and you must reconnect. The default timeout is five minutes. You can set the timeout and other session-specific options using the `line-vty` command.

Syntax

```
#(config) line-vty
```

This changes the prompt to:

```
#(config line-vty)
```

Subcommands

`#(config line-vty)` **`exit`**
Exits configure line-vty mode and returns to configure mode.

`#(config line-vty)` **`length`** *`num_lines_on_screen`*
Specifies the number of lines of code that should appear on the screen at one time. Specify 0 to scroll without pausing.

`#(config line-vty)` **`no length`**
Disables screen paging.

`#(config line-vty)` **`telnet`** {**`no transparent`** | **`transparent`**}
Indicates that this is a Telnet protocol-specific configuration. If you specify `no transparent`, carriage returns are sent to the console as a carriage return plus linefeed. If you specify `transparent`, carriage returns are sent to the console as a carriage return.

`#(config line-vty)` **`timeout`** *`minutes`*
Sets the line timeout to the number of minutes indicated by *`minutes`*.

`#(config line-vty)` **`view`**
Displays running system information.

`#(config line-vty)` **`width`**
Sets the width of the display terminal.

Example

```
SGOS#(config) line-vty
SGOS#(config line-vty) timeout 60
 ok
SGOS#(config line-vty) exit
SGOS#(config)
```

#(config) load

See

- **# load** on page 62

#(config) management-services

Synopsis

The ProxySG provides the following console services:

- HTTP (Not enabled by default)
- HTTPS
- SSH
- Telnet (Not created by default; a Telnet proxy service is created by default on port 23.)

The ProxySG also provides SNMP management services.

Syntax

```
#(config) management-services
```

This changes the prompt to:

```
#(config management-services)
```

Subcommands

The options below allow you to manage the console service.

`#(config management-services)` **`create`** {**`http-console`** *`service_name`* | **`https-console`** *`service_name`* | **`ssh-console`** *`service_name`* | **`telnet-console`** *`service_name`* | **`snmp`** *`service_name`*}
Creates a console or SNMP service with the service name you choose.

`#(config management-services)` **`delete`** *`service_name`*
Deletes the specified console name or SNMP service name.

`#(config management-services)` **`edit`** *`service_name`*
Changes the prompt, depending on the console or SNMP service you choose:

- **`#(config http-console)`** on page 240
- **`#(config https-console)`** on page 241
- **`#(config ssh-console)`** on page 243
- **`#(config telnet-console)`** on page 244
- **`#(config snmp_service_name)`** on page 245

`#(config management-services)` **`exit`**
Leaves management-services submode; returns to the config prompt.

`#(config management-services)` **`view`**
Views all console services.

Note: If you create a console name with spaces, the name must be enclosed in quotes; for example, "My Console1".

#(config http-console)

Synopsis

This console service intercepts HTTP traffic, usually on port 80. This console service is created but not enabled due to security concerns.

Syntax

```
#(config management-services) edit http_console
```

This changes the prompt to:

```
#(config http_console)
```

Subcommands

`#(config http_console) add {all | proxy_ip_address} port {enable | disable}`
Add a listener to the console service. *All* selects all IPv4 and IPv6 addresses on the proxy; alternatively, you can select a specific proxy's IPv4/IPv6 address. When specifying IPv6 addresses, only global (not linklocal) addresses can be used. You must always choose a port. By default the listener is enabled.

`#(config http_console) disable {all | proxy_ip_address} port`
Disables the specified listener.

`#(config http_console) enable {all | proxy_ip_address} port`
Enables the specified listener.

`#(config http_console) exit`
Exits to the (config management-services) prompt.

`#(config http_console) remove {all | <proxy-ip> <port>}`
Removes the specified listener(s).

`#(config http_console) view`
Views a summary of the console service's configuration.

For More Information

- **#(config) management-services** on page 239
- *SGOS Administration Guide*

Example

```
SGOS#(config) management-services
SGOS#(config management-services) create http-console http_console
SGOS#(config management-services) edit http_console
SGOS#(config http_console) add 10.25.36.47 80
SGOS#(config http_console) enable 10.25.36.47 80
```

#(config https-console)

Synopsis

The HTTPS console intercepts traffic on ports 8082. You can create additional HTTPS consoles if necessary.

Syntax

```
#(config management-services) edit https_console
```

This changes the prompt to:

```
#(config https_console)
```

Subcommands

`#(config https_console)` **`add`** {**`all`** | *`proxy_ip_address`*} *`port`* {**`enable`** | **`disable`**}
Add a listener to the console service. *`All`* selects all IPv4 and IPv6 addresses on the proxy; alternatively, you can select a specific proxy's IPv4/IPv6 address. When specifying IPv6 addresses, only global (not linklocal) addresses can be used. You must always choose a port. By default the listener is enabled.

`#(config https_console)` **`attribute cipher-suite`** [<*`cipher-suite`*>]+
Associates one or more ciphers with the console service. A Cipher suite can be any combination of the following:

`AES128-SHA256`
`AES256-SHA256`
`AES128-SHA`
`AES256-SHA`
`DHE-RSA-AES128-SHA`
`DHE-RSA-AES256-SHA`
`DES-CBC3-SHA`
`RC4-SHA`
`RC4-MD5`
`DES-CBC-SHA`
`EXP-DES-CBC-SHA`
`EXP-RC4-MDS`
`EXP-RC2-CBC-MD5`

`#(config https_console)` **`attribute keyring`** *`keyring_ID`*
Specifies the keyring ID you want to use with this console.

`#(config https_console)` **`attribute ssl-versions`** {**`sslv2`** | **`sslv3`** | **`tlsv1`**| **`tlsv1.1`**| **`tlsv1.2`**}
Selects the SSL versions to use**.**

`#(config https_console)` **`disable`** {**`all`** | *`proxy_ip_address`*} *`port`*
Disables the specified listener.

`#(config https_console)` **`enable`** {**`all`** | *`proxy_ip_address`*} *`port`*
Enables the specified listener.

`#(config https_console)` **`exit`**
Exits to the (config management-services) prompt.

`#(config https_console)` **`remove`** {**`all`** | <*`proxy-ip`*> <*`port`*>}
Removes the specified listener(s).

`#(config https_console)` **`view`**
Views a summary of the console service's configuration.

For More Information

- **`#(config) management-services`** on page 239
- *SGOS Administration Guide*

Example

```
SGOS#(config) management-services
SGOS#(config management-services) create https-console https_console
SGOS#(config management-services) edit https_console
SGOS#(config https_console) add 10.25.36.47 80
SGOS#(config https_console) enable 10.25.36.47 80
SGOS#(config https_console) attribute cipher-suite rc4-md5 des-cbc-sha
aes128-sha
```

Note: For a discussion of available ciphers, refer to *SGOS Administration Guide*, Managing the ProxySG chapter

#(config ssh-console)

Synopsis

The SSH console service allows to you to securely connect to the Command Line Interface. By default, SSHv2 is enabled and assigned to port 22. You do not need to create a new host key unless you want to change the existing configuration.

To manage new host keypairs or global settings for all SSH console services, use the `#(config) ssh-console` command. For more information, see **`#(config) ssh-console`** on page 398.

Syntax

`#(config management-services)`**`create ssh-console`** *`ssh_console_name`*

`#(config management-services)`**`edit`** *`ssh_console_name`*

This changes the prompt to:

`#(config` *`ssh_console_name`*`)`

Subcommands

`#(config` *`ssh_console_name`*`)` **`add`** {**`all`** | *`proxy_ip_address`*} *`port`* {**`enable`** | **`disable`**}
Add a listener to the console service. *`All`* selects all IPv4 and IPv6 addresses on the proxy; alternatively, you can select a specific proxy's IPv4/IPv6 address. When specifying IPv6 addresses, only global (not linklocal) addresses can be used. You must always choose a port. By default the listener is enabled.

`#(config` *`ssh_console_name`*`)` **`disable`** {**`all`** | *`proxy_ip_address`*} *`port`*
Disables the specified listener.

`#(config` *`ssh_console_name`*`)` **`enable`** {**`all`** | *`proxy_ip_address`*} *`port`*
Enables the specified listener.

`#(config` *`ssh_console_name`*`)` **`exit`**
Exits to the (config management-services) prompt.

`#(config` *`ssh_console`*`)` **`remove`** {**`all`** | *`<proxy-ip>`* *`<port>`*}
Removes the specified listener(s).

`#(config` *`ssh_console_name`*`)` **`view`**
Views a summary of the console service's configuration.

For More Information

- **`#(config) management-services`** on page 239
- **`#(config) ssh-console`** on page 398

Example

```
SGOS#(config) ssh-console
SGOS#(config ssh-console) create host-keypair
SGOS#(config management-services) edit ssh_console
SGOS#(config ssh_console) add 10.25.36.47 80
SGOS#(config ssh_console) enable 10.25.36.47 80
```

#(config telnet-console)

Synopsis

This console service provides access to the administrative CLI through Telnet. Due to security concerns, use of this console is not recommended.

A shell Telnet proxy service is created on port 23. If you do decide to create a Telnet console, you must first remove the Telnet proxy service and apply the changes. You can later re-add the Telnet proxy service on a different port.

Syntax

```
#(config management-services) edit telnet_console
```

This changes the prompt to:

```
#(config telnet_console)
```

Subcommands

`#(config telnet_console) add {all | proxy_ip_address} port {enable | disable}`
Add a listener to the console service. *All* selects all IPv4 and IPv6 addresses on the proxy; alternatively, you can select a specific proxy's IPv4/IPv6 address. When specifying IPv6 addresses, only global (not linklocal) addresses can be used. You must always choose a port. By default the listener is enabled.

`#(config telnet_console) disable {all | proxy_ip_address} port`
Disables the specified listener.

`#(config telnet_console) enable {all | proxy_ip_address} port`
Enables the specified listener.

`#(config telnet_console) exit`
Exits to the (config management-services) prompt.

`#(config telnet_console) remove {all | <proxy-ip> <port>}`
Removes the specified listener(s).

`#(config telnet_console) view`
Views a summary of the console service's configuration.

For More Information

- **#(config) management-services** on page 239
- *SGOS Administration Guide*

Example

```
SGOS#(config) management-services
SGOS#(config management-services) create telnet-console telnet_console
SGOS#(config management-services) edit telnet_console
SGOS#(config telnet_console) add 10.25.36.47 80
SGOS#(config telnet_console) enable 10.25.36.47 80
```

#(config snmp_service_name)

Synopsis

The SNMP management service provides an explicit connection for communicating with the ProxySG. You can create an SNMP listener for any available port and for all available ProxySG IP addresses or for a specific IPv4 or IPv6 address.

Syntax

```
#(config management-services) edit snmp_service_name
```

This changes the prompt to:

```
#(config snmp_service_name)
```

Subcommands

`#(config snmp_service_name) add {all|<proxy-ip> <port> {enable|disable}`
Add an SNMP listener to the management service. `All` selects all IPv4 and IPv6 addresses on the proxy; alternatively, you can select a specific proxy's IPv4/IPv6 address. You must always choose a port. By default, the listener is enabled.

`#(config snmp_service_name) disable {all|<proxy-ip> <port>}`
Disable a specific SNMP listener.

`#(config snmp_service_name) enable {all|<proxy-ip> <port>}`
Enable a specific SNMP listener.

`#(config snmp_service_name) exit`
Return to the `(config management-services)` prompt.

`#(config snmp_service_name) remove {all | <proxy-ip> <port>}`
Remove an SNMP listener.

`#(config snmp_service_name) view`
Show the SNMP listener configuration.

For More Information

- **#(config) management-services** on page 239
- **#(config) snmp** on page 384
- *SGOS Administration Guide*

Example

```
SGOS#(config) management-services
SGOS#(config management-services) create snmp mysnmp
     ok
SGOS#(config management-services) edit mysnmp
SGOS#(config mysnmp) view
   Service name:    my-snmp
   Service:         SNMP
   Destination IP   Port Range
   <All>            161  Enabled
```

#(config) mapi

Synopsis

Configures MAPI.

Syntax

```
SGOS#(config) mapi
```

This changes the prompt to:

```
SGOS#(config mapi) [subcommands]
```

Subcommands

`SGOS#(config mapi) batching {enable | disable}`
Enables or disables batching. The default is enabled.

`SGOS#(config mapi) cas-virtual-ip ip_address`
Configures the virtual IPv4 address of a Client Access Server (CAS) array so that the ProxySG can create a new listener for this VIP. After this is configured, the MAPI connections to the CAS array virtual host can be intercepted and optimized. This setting must be configured on each Branch peer that will handle MAPI traffic to an Exchange server with a third party load balancer, and must be set before Outlook connects to the Exchange Server. Only one VIP can be configured per ProxySG.

`SGOS#(config mapi) encrypted-acceleration {enable | disable}`
Enables or disables acceleration of encrypted MAPI. The default is enabled.

`SGOS#(config mapi) exchange-domain domain_name_alias`
Selects the MAPI exchange domain name alias to use. This command is required for accelerating encrypted MAPI.

`SGOS#(config mapi) exit`
Exits the MAPI mode and returns to SGOS#(config) mode.

`SGOS#(config mapi) handoff (enable | disable}`
Use the endpoint-mapper service. The default is enabled.

`SGOS#(config mapi) keep-alive duration 1-168`
Sets the length of time, in hours, that the session is active. The default is 72 hours.

`SGOS#(config mapi) keep-alive {enable | disable}`
Enables the keep-alive configuration. The default is disabled.

`SGOS#(config mapi) keep-alive interval 15-60`
Sets the length of time, in minutes, before the service checks for new e-mail. The default is 30 minutes.

`SGOS#(config mapi) keep-alive max-sessions 1-200`
Sets the maximum number of active sessions at any given point. The default is 100 sessions. If the limit is reached, the oldest session is dropped.

`SGOS#(config mapi) no exchange-domain | cas-virtual-ip`
Clears the settings for Exchange domain alias or CAS virtual IP address.

`SGOS#(config mapi) view`
Views the MAPI configuration.

For More Information

- **#(config Endpoint Mapper)** on page 284

Example

```
SGOS#(config mapi) view
Batching:                                enabled
Keep-Alive:                              disabled
Keep-Alive Duration (hours):             72
Keep-Alive Interval (minutes):           30
Keep-Alive Maximum Sessions:             100
Endpoint Mapper Handoff:                  enabled
```

#(config) netbios

Synopsis

Use this command to configure NetBIOS.

Syntax

```
#(config) netbios
```

This changes the prompt to:

```
#(config netbios)
```

Subcommands

`#(config netbios) exit`
Exits configure netbios mode and returns to configure mode.

`#(config netbios) nbstat {requester {retries | timeout} | responder {enable | disable}}`
Requester is enabled by default and cannot be disabled, with three retries and a five-second timeout. Responder is disabled by default. Note that the requestor is used only if you write policy that uses it. If no features that trigger a request are used, no traffic is sent.

`#(config netbios) view`
Shows the NetBIOS settings.

Example

```
SGOS#(config) netbios
SGOS#(config netbios) nbstat responder enable
 ok
SGOS#(config netbios) exit
SGOS#(config)
 ok
```

#(config) netflow

Synopsis

NetFlow is a network protocol developed by Cisco Systems to monitor and export IP traffic information. Use the `#(config)netflow` command to configure NetFlow. After you configure NetFlow on the appliance, direct the flow data to the collectors that you have already set up.

If you enable NetFlow on the ProxySG appliance (it is disabled by default), it observes network flows on all interfaces and keeps track of flow statistics, such as source and destination IP addresses, the size of the flow (in terms of packets and bytes), and when the flow was sent. After the appliance gathers the flow statistics, it exports them in NetFlow records to a remote system called a collector, such as Blue Coat IntelligenceCenter.

Important: Because NetFlow runs on UDP, the ProxySG appliance cannot verify collector configuration. You should make sure that collector IP address and port are correct before setting up NetFlow in the CLI.

Currently, SGOS supports NetFlow v5, which is restricted to collecting flow statistics for IPv4 packets only.

Terminology

Blue Coat documentation uses the following terms to describe the NetFlow feature in SGOS.

- *Network flow*—A sequence of packets from a source application to a destination application. A network flow has attributes such as IP address, port, protocol, and inbound/outbound interfaces.

  A flow is exported to the collectors when:

  - it is reported as being finished (for example, traffic for an existing flow stops)
  - it has been inactive for a period of time exceeding the `inactive-timeout` value (for example, the connection is stale)
  - it has been active for a period of time exceeding the `active-timeout` value (for example, it is a long-running flow)
  - it exceeds the record byte count limit of 2^32 bytes

- *Flow records*—Contain information about a flow, such as source and destination IP addresses, the amount of data transferred (in terms of packets and bytes), and the flow start and end times.
- *Inbound/outbound interfaces*—Flow records sent to the ProxySG appliance are exported on inbound interfaces. Flow records originating from the ProxySG appliance are exported on outbound interfaces.
- *NetFlow packets*—NetFlow-formatted packets, which contain copies of expired flows. These packets are sent to a collector once they reach the maximum of 30 records, or two minutes after the first flow record is collected.

Syntax

```
#(config) netflow
```

This enters NetFlow mode and changes the prompt to:

```
#(config netflow)
```

Subcommands

`#(config netflow)` **`active-timeout`** *`timeout-seconds`*
Specifies the age of an active flow, after which it is reported. When an active flow exceeds the maximum time, the flow is reported containing the flow statistics up to that point. The default is 1800 seconds.

Note: The `active-timeout` value must be greater than the `inactive-timeout` value.

`#(config netflow)` **`collectors`**
Configures NetFlow collector(s). See **`#(config netflow) collectors`** on page 251.

`#(config netflow)` **`disable`**
Sends the remaining flow data to the collectors and disables the capture of any more Netflow information. This is the default setting.

`#(config netflow)` **`enable`**
Enables the appliance to begin monitoring if one or more collectors have been configured. See **`#(config netflow) collectors`** on page 251.
If no collectors are defined when you enter this command, the CLI warns you to configure at least one collector.

`#(config netflow)` **`exit`**
Exits NetFlow mode.

`#(config netflow)` **`inactive-timeout`** *`timeout-seconds`*
Specifies the maximum amount of time a flow is considered active without seeing network traffic. When the maximum is exceeded, the appliance determines that the flow is inactive and exports a flow record. The default is 15 seconds.

Note: The inactive-timeout value must be less than the active-timeout value.

`#(config netflow)` **`interfaces`**
Configures NetFlow interfaces. See **`#(config netflow) interfaces`** on page 253.

#(config netflow) **`view`**
Displays NetFlow configuration settings and statistics:

- NetFlow state
- Number of active flows
- Collector information
- Active timeout value
- Inactive timeout value

The `#show netflow` command also displays these settings and statistics.

Example

Enable the NetFlow feature and exit the netflow subnode.

```
#(config) netflow
#(config netflow) enable
  ok
#(config netflow) exit
#(config)
  ok
```

#(config netflow) collectors

Synopsis

Use this command to configure NetFlow collectors. A NetFlow collector is a software application, such as ManageEngine® NetFlow Analyzer, that accumulates the data from the ProxySG appliance, which acts as the flow record exporter.

You define a collector uniquely by both address and port. This allows you to configure multiple collectors on different ports for the same machine.

You can configure up to four collectors to collect the flow records from the ProxySG appliance. Note that collectors on the same machine are counted separately.

Important: The appliance exports NetFlow records over UDP, which does not guarantee that the data will be sent to a destination; thus, the appliance cannot verify if collector configuration is correct. Blue Coat recommends that you ensure collectors are configured correctly before setting up NetFlow. Then, when you add a collector in the CLI, be sure to enter the correct IP address and port.

In addition, because UDP does not attempt to re-send lost data, configuring more than collector can help establish some redundancy in your NetFlow setup. For example, consider a NetFlow setup with two collectors. The NetFlow data streams sent to both collectors may lose different packets and different amounts of data, but you can inspect both sets of partial data to gain a more complete picture of those specific flows.

Syntax

```
#(config) netflow
```

```
#(config netflow) collectors
```

This changes the prompt to:

```
#(config netflow collectors)
```

Subcommands

`#(config netflow collectors) add` *IP-address port*

Adds a collector on either an IPv4 or IPv6 address. Specify the collector's IP address and the port on which it is listening. You can add multiple collectors on different ports on the same machine. If NetFlow is disabled, adding collectors does not enable NetFlow. To enable NetFlow, use the `#(config netflow) enable` command.

Note: If you specify IPv4 addresses to configure collectors, you must use unique IP addresses and ports. If you use different strings to specify the same logical IP address (for example, canonical and abbreviated forms for the same address), the ProxySG appliance detects the duplicate IP address.

`#(config netflow collectors) clear`

Clears the list of all configured collectors.

`#(config netflow collectors) exit`

Exits collector mode.

`#(config netflow collectors) remove` *IP-address port*

Removes a collector from the list. Specify the collector's IPv4 or IPv6 IP address and the port on which it is listening. If NetFlow is enabled and you remove all collectors, a warning message appears and NetFlow collection will be suspended, although the feature remains enabled.

`#(config netflow collectors) view`

Displays the list of configured collectors.

Example

Enter NetFlow collector mode, add a collector, and exit NetFlow collector mode.

```
#(config) netflow
#(config netflow) collectors
#(config netflow collectors)
#(config netflow collectors) add 192.0.2.0 9800
  ok
#(config netflow collectors) exit
```

#(config netflow) interfaces

Synopsis

Use this command to configure NetFlow interfaces.

Syntax

Enter NetFlow interfaces mode.

```
#(config) netflow
#(config netflow) interfaces
#(config netflow interfaces)
```

Subcommands

- `#(config netflow interfaces) add all`

  Adds all interfaces for NetFlow processing.

- `#(config netflow interfaces) add <adaptor>:<interface> [in|out|inout]`

  Adds an interface used for processing NetFlow input (in), output (out), or both (inout). If no parameter is specified, the default is used (inout).

Note: By default, all interfaces are included for NetFlow processing. You need only add interfaces that you removed previously.

- `#(config netflow interfaces) remove all`

  Removes all interfaces for NetFlow processing.

- `#(config netflow interfaces) remove <adaptor>:<interface> [in|out|inout]`

  Removes the interface used for processing NetFlow input (in), output (out), or both (inout). If no parameter is specified, the default is used (inout).

  If you remove all interfaces from NetFlow processing, and NetFlow is enabled, the CLI warns you that no interfaces are available for processing and that collection will be disabled.

Example

Add an interface, which was removed previously, for processing NetFlow input and output.

```
#(config netflow interfaces)add 1:0 inout
  ok
```

#(config) no

Synopsis

Use this command to negate the current settings for the archive configuration, content priority, IP default gateway, SOCKS machine, or system upgrade path.

Syntax

`#(config)` **`no archive-configuration`**
Clears the archive configuration upload site.

`#(config)` **`no bridge`** *`bridge_name`*
Clears the bridge configuration.

`#(config)` **`no content`** {**`priority`** {**`regex`** *`regex`* | **`url`** *`url`*} | **`outstanding-requests`** {**`delete`** *`regex`* | **`priority`** *`regex`* | **`revalidate`** *`regex`*}}
`priority` {regex *`regex`* | url *`url:`* Removes a deletion regular expression policy or a deletion URL policy.
`outstanding-requests` {*`delete`* | *`priority`* | *`revalidate`*} *`regex:`* Deletes a specific, regular expression command in-progress (revalidation, priority, or deletion).

`#(config)` **`no ip-default-gateway`** *`ip_address`*
Sets the default gateway IP address to zero.

`#(config)` **`no socks-machine-id`**
Removes the SOCKS machine ID from the configuration.

`#(config)` **`no ui-update-path`**
Clears the UI update path.

`#(config)` **`no upgrade-path`**
Clears the upgrade image download path.

For More information

- *SGOS Administration Guide*

Example

```
SGOS#(config) no archive-configuration
 ok
SGOS#(config) no content priority regex http://.*cnn.com
 ok
SGOS#(config) no content priority url http://www.bluecoat.com
 ok
SGOS#(config) no ip-default-gateway 10.252.10.50
 ok
SGOS#(config) no socks-machine-id
 ok
SGOS#(config) no upgrade-path
 ok
```

#(config) ntp

Synopsis

Use this command to set NTP parameters. Network Time Protocol (NTP) is a protocol that is used to synchronize computer clock times in a network of computers. The ProxySG sets the UTC time by connecting to an NTP server. The ProxySG includes a list of NTP servers available on the Internet. If an NTP server is not available, you can set the time manually using the Management Console.

Syntax

`#(config) ntp clear`
Removes all entries from the NTP server list.

`#(config) ntp {enable | disable}`
Enables or disables NTP.

`#(config) ntp interval` *`minutes`*
Specifies how often to perform NTP server queries.

`#(config) ntp no server` *`domain_name`*
Removes the NTP server named *`domain_name`* from the NTP server list.

`#(config) ntp server` *`domain_name`*
Adds a server to the NTP server list. Enter either a domain name of an NTP server that resolves to an IPv4 or IPv6 address, or an IPv4 or IPv6 address of an NTP server.

For More Information

- *SGOS Administration Guide*

Example

```
SGOS#(config) ntp server clock.tricity.wsu.edu
 ok
```

#(config) policy

Synopsis

Use this command to specify central and local policy file location, status, and other options.

Syntax

`#(config)` **`policy central-path`** *`url`*
Specifies the network path (indicated by *`url`*) from which the central policy file can be downloaded.

`#(config)` **`policy forward-path`** *`url`*
Specifies the network path (indicated by *`url`*) from which the forward policy file can be downloaded.

`#(config)` **`policy hmac encrypted-key`** *`encrypted-key`*
Sets the encrypted key used by the HMAC (Hash-based Message Authentication Code) policy substitution function to perform digital signatures.
Note: If you administer multiple ProxySG appliances using Director, use this command to synchronize the HMAC secret key across all appliances and to restore the key when an appliance is remanufactured.

`#(config)` **`policy hmac generate-key`**
Generates a new random key to use for `:hmac` policy substitution function to perform digital signatures.

`#(config)` **`policy local-path`** *`url`*
Specifies the network path (indicated by *`url`*) from which the local policy file can be downloaded.

`#(config)` **`policy no central-path`**
Specifies that the current central policy file URL setting should be cleared.

`#(config)` **`policy no forward-path`**
Specifies that the current forward policy file URL setting should be cleared.

`#(config)` **`policy no local-path`**
Specifies that the current local policy file URL setting should be cleared.

`#(config)` **`policy no notif`**y
Specifies that no e-mail notification should be sent if the central policy file should change.

`#(config)` **`policy no subscribe`**
Specifies that the current policy should not be automatically updated in the event of a central policy change.

`#(config)` **`policy no vpm-cpl-path`**
Clears the network path to download VPM CPL policy.

`#(config)` **`policy no vpm-xml-path`**
Clears the network path to download VPM XML policy.

`#(config)` **`policy notify`**
Specifies that an e-mail notification should be sent if the central policy file should change.

`#(config)` **`policy order`** *`order of v)pm, l)ocal, c)entral`*
Specifies the policy evaluation order.

`#(config)` **`policy poll-interval`** *`minutes`*
Specifies the number of minutes that should pass between tests for central policy file changes.

`#(config)` **`policy poll-now`**
Tests for central policy file changes immediately.

`#(config)` **`policy proxy-default`** {**`allow`** | **`deny`**}
allow: The default proxy policy is `allow`.
deny: The default proxy policy is `deny`.

`#(config)` **`policy reset`**
Clears all policies.

`#(config)` **`policy subscribe`**
Indicates that the current policy should be automatically updated in the event of a central policy change.

`#(config)` **`policy vpm-cpl-path`** *`url`*
Specifies the network path (indicated by *`url`*) from which the vpm-cpl policy file can be downloaded.

`#(config)` **`policy vpm-xml-path`** *`url`*
Specifies the network path (indicated by *`url`*) from which the vpm-xml policy file can be downloaded.

For More Information

- *SGOS 6.x Visual Policy Manager Reference*

Example

```
SGOS#(config) policy local-path http://www.server1.com/local.txt
 ok
SGOS#(config) policy central-path http://www.server2.com/central.txt
 ok
SGOS#(config) policy poll-interval 10
```

#(config) private-network

Synopsis

Allows you to configure information on the private network(s) in your environment.

```
SGOS#(config) private-network
```

This changes the prompt to:

```
SGOS#(config private-network) [subcommands]
```

Subcommands

SGOS#(config private-network) add {**subnet** *<subnet_prefix>* [*/<prefix_length>*] | **domain** *domain_name*}
Allows you to add specific private network subnets or domains.

SGOS#(config private-network) clear-all {**subnets** | **domains**}
Clears or removes all private network subnets and domains.

SGOS#(config private-network) exit
Exits the private network configuration and brings you back to the configuration prompt.

SGOS#(config private-network) remove {**subnet** *<subnet_prefix>* [*/<prefix_length>*] | **domain** *domain_name*}
Allows you remove specific private network subnets or domains.

SGOS#(config private-network) restore-non-routable-subnets
Restores the default non-routable subnets to the private network configuration.

SGOS#(config private-network) view
View configured private netwoirks and domains.

For More Information

- *SGOS Administration Guide*

Example

```
SGOS#(config private-network) add 1.2.3.4
SGOS#(config private-network) add 1.2.0.0/16
SGOS#(config private-network) remove domain bluecoat.com
```

#(config) profile

Synopsis

Sets your system profile to normal (the default setting) or portal (to accelerate the server).

Syntax

`#(config) profile bwgain`
Sets your system profile to bandwidth gain.

`#(config) profile normal`
Sets your system profile to normal.

`#(config) profile portal`
Sets your system profile to portal.

For More Information

- *SGOS Administration Guide*

Example

```
SGOS#(config) profile normal
 ok
```

#(config) proxy-client

Synopsis

Use this command to configure the Client Manager and client configuration options for the ProxyClient. Replaces the earlier `#(config) sg-client` command.

Syntax

`#(config) sg-client`

or

`#(config) proxy-client`

This changes the prompt to:

`#(config proxy-client)`

Subcommands

`#(config proxy-client) acceleration`
Changes the prompt to **#(config proxy-client acceleration)** on page 263

`#(config proxy-client) clear {inactive | all}`
Clears (that is, sets to zero) the count of inactive ProxyClients or all ProxyClients.

Clients are automatically cleared after 30 days of inactivity.

After a software upgrade, clients appear twice for 30 days—one entry for the earlier version of client software and one entry and one entry for the newer version of client software. You can optionally clear the inactive clients to avoid seeing duplicate information.

For a client to be reported as inactive, 10 minutes or more must elapse between heartbeat packets it sends to the Client Manager.

`#(config proxy-client) enable`
Enable this appliance as the Client Manager. You can have only one Client Manager in your ADN network.

`#(config proxy-client) disable`
Do not use this appliance as the Client Manager.

`#(config proxy-client) client-manager host {from-client-address | <ip-address | host>}`
Identify this appliance as the Client Manager in one of the following ways:

- `from-client-address`: (*Recommended*.) Use this command if you want clients to download the ProxyClient software, configuration, and updates from the host from which the clients originally obtained the software.
- *`ip-address`* or *`host`*: Use this command only if you want to change the host from which clients download the ProxyClient software, configuration, and updates. Enter a fully-qualified host name or IP address only; do not preface the with `http://` or `https://`or downloads will fail.

  In other words, this option enables you to change the host from which currently-installed clients obtain future software and configuration updates. Use caution when selecting this option because if clients are unable to connect to the host you enter in the adjacent field, new installations from the Client Manager and updates to existing installations will fail.

Note: Blue Coat recommends you enter the fully-qualified host name. If you enter either an unqualified host name or IP address and change it later, connections to all currently-connected clients are dropped.

`#(config proxy-client)` **`client-manager install-port`** *`port`*
Port on which the host you entered in the preceding option listens for requests from clients. The default is 8084.

`#(config proxy-client)` **`client-manager keyring`** *`keyring`*
Name of the keyring the Client Manager will use when clients connect to it.

`#(config proxy-client)` **`exit`**
Exits the proxy client configuration prompt and brings you back to the configuration prompt.

`#(config proxy-client)` **`locations`**
Changes the prompt to **`#(config proxy-client locations)`** on page 268

`#(config proxy-client)` **`software-upgrade-path`** *`url`*
Sets the URL used to upload updated ProxyClient software to the Client Manager so it can make the latest ProxyClient software available to update or to install on client machines.

Important: After you update the Client Manager, whenever users connect using the ProxyClient, they will be required to update the ProxyClient software.

Upload the ProxyClient software from a URL in the following format:

`https://`*`host`*`:`*`port`*`/`*`path`*`/ProxyClient.car`

For example,

`https://myapache.example.com:8088/images/ProxyClient.car`

After you set the path from which to load the updates, see **`# load`** on page 62.

`#(config proxy-client) [no]` **`uninstall-password`** [*`password`*]
Enter a plain text password that is required if users want to uninstall the ProxyClient software.

Do any of the following to remove any previous uninstall password:

- Enter **`uninstall-password`** without an argument.
- Enter **`no uninstall-password`**.

`#(config proxy-client)` **`hashed-uninstall-password`** [*`hashed-password`*]
Enter the uninstall password hashed by the Blowfish algorithm. You can use the hashed password in scripts when you do not want to expose the password in plain text. The only way to know the Blowfish-hashed password is to view it using the **`show-config`** command.

The hashed password displays as follows:

```
proxy-client ;mode
hashed-uninstall-password
"$2a$05$XyjVSFGvPkTmUi6zKDmyauSArzwka62evn7c13k6qUenR.KOEz4IC"
```

`#(config proxy-client)` **`update-interval`** *`minutes`*
Frequency clients check with the Client Manager for updated ProxyClient software. Valid values are 10-432000 (that is, 300 days). Default is 120.

`#(config proxy-client)` **`view`**
View current Client Manager settings.

`#(config proxy-client)` **`web-filtering`**
Changes the prompt to **`#(config proxy-client web-filtering)`** on page 271

For More Information

- *ProxyClient Administration and Deployment Guide*

Example

```
SGOS#(config) client-manager host enable

SGOS#(config) client-manager host from-client-address

SGOS#(config) software-upgrade-path
   https://myapache.example.com:8088/images/ProxyClient.car
```

#(config proxy-client acceleration)

Synopsis

Configure acceleration settings for ProxyClients.

Syntax

```
#(config) sg-client
```

or

```
#(config) proxy-client
```

This changes the prompt to:

```
#(config proxy-client)
```

Enter

```
#(config proxy-client) acceleration
```

This changes the prompt to:

```
#(config proxy-client acceleration)
```

Subcommands

`#(config proxy-client acceleration)` **`adn`**
Change to `acceleration adn` mode. For more information, see **`#(config proxy-client acceleration adn)`** on page 264.

`#(config proxy-client acceleration)` **`cifs`**
Change to `acceleration cifs` mode. For more information, see **`#(config proxy-client acceleration cifs)`** on page 266.

`#(config proxy-client acceleration)` **`disable`**
Disables all acceleration for ProxyClients; that is, gzip compression, CIFS protocol optimization, and byte caching.

`#(config proxy-client acceleration)` **`enable`**
Enables acceleration for ProxyClients; that is, gzip compression, CIFS protocol optimization, and byte caching.

`#(config proxy-client acceleration)` **`exit`**
Exits `acceleration` submode and returns to `proxy-client` mode.

`#(config proxy-client acceleration)` **`max-cache-disk-percent`** *`percentage`*
Maximum percentage of client disk space to use for caching objects, such as CIFS objects. Valid values are 10—90; default is 10.

`#(config proxy-client acceleration)` **`view`**
Displays current ProxyClient acceleration settings.

For More Information

- *ProxyClient Administration and Deployment Guide*

Example

```
SGOS#(config proxy-client acceleration) max-cache-disk-percent 15
SGOS#(config proxy-client acceleration) enable
```

#(config proxy-client acceleration adn)

Synopsis

Configure ADN manager and ADN rules settings for ProxyClients.

Syntax

```
#(config) sg-client
```

or

```
#(config) proxy-client
```

This changes the prompt to:

```
#(config proxy-client)
#(config proxy-client) adn
```

This changes the prompt to:

```
#(config proxy-client acceleration adn)
```

Subcommands

`#(config proxy-client acceleration adn)` **`primary-manager`** *`ip-address`*
The IP address of the primary ADN manager. The ADN manager keeps track of and advertises the routes of the appliances it knows about. You must specify a primary manager.

The ProxyClient obtains the routing table from the ADN manager.

`#(config proxy-client acceleration adn)` **`backup-manager`** *`ip-address`*
The IP address of the backup ADN manager. Configuring a backup ADN manager is optional but recommended.

If the ADN manager becomes unavailable for any reason, the backup ADN manager takes over the task of advertising routes to all ADN nodes, such as the ProxyClient.

`#(config proxy-client acceleration adn)` **`manager-port`** *`port`*
ADN manager and backup manager plain listen port. (To use the ProxyClient in your ADN network, the ADN manager's listening mode must be configured for **`plain-only`**, **`secure-only`**, or **`both`**. For more information, see **`#(config) adn`** on page 109.

`#(config proxy-client acceleration adn)` **`port-list`** {**`exclude-ports`** | **`include-ports`**}
Determines whether you will use the include ports list or exclude ports list.

`#(config proxy-client acceleration adn)` {**`exclude-ports`** *`port list,port-range`* | **`include-ports`** *`port list,port-range`*}
Determines which TCP ports to exclude or include in ADN tunnels. Assuming clients using the ProxyClient software can connect to an ADN peer that can optimize traffic to the destination IP address, this setting determines ports the clients can use (or not use).

For example, you can exclude ports or port ranges because traffic coming from those ports has already been encrypted.

For example, the following command excludes traffic from ports 22 and 443 from being routed through ADN:

```
#(config proxy-client acceleration adn) exclude-ports 22,443
```

Valid values: Comma-separated list of ports and port ranges (no spaces, separated by a dash character: *<port>*,*<port>*,*<port>*-*<port>*.

`#(config proxy-client acceleration adn)` **`exclude-subnets`**
Configure the subnets excluded from ADN acceleration.

`#(config proxy-client acceleration adn exclude-subnets)` **{add | remove}** *subnet_prefix[/prefix length]*
Adds or removes subnets from the excluded subnets list, which is the list of subnets not included in ADN tunnels. Use a comma-separated list of IP addresses and subnets in CIDR notation.

For example, the following command excludes traffic from IP address 192.168.0.1—192.168.0.254 from being routed through an ADN tunnel:

```
#(config proxy-client acceleration adn exclude-subnets) add 192.168.0.1/24
```

`#(config proxy-client acceleration adn exclude-subnets)` **clear**
Removes all subnets from the current excluded subnet list. In other words, traffic from all IP addresses and subnets will be routed through the ADN tunnel.

`#(config proxy-client acceleration adn exclude-subnets)` **exit**
Exits the `exclude-subnets` submode.

`#(config proxy-client acceleration adn exclude-subnets)` **view**
View the list of excluded subnets.

`#(config proxy-client acceleration adn)` **no** {**primary-manager** | **backup-manager**}
Clears the backup or primary ADN manager IP address

`#(config proxy-client acceleration adn)` **exit**
Exit the `adn` submode.

For More Information

- *ProxyClient Administration and Deployment Guide*

Example

```
#(config proxy-client acceleration adn) exclude-ports
22,88,443,993,995,1352,1494,1677,3389,5900
#(config proxy-client acceleration adn) primary-manager 198.162.0.10
```

#(config proxy-client acceleration cifs)

Synopsis

Configure CIFS settings for ProxyClients.

Syntax

```
#(config) sg-client
```

or

```
#(config) proxy-client
```

This changes the prompt to:

```
#(config proxy-client)
#(config proxy-client) cifs
```

This changes the prompt to:

```
#(config proxy-client acceleration cifs)
```

Subcommands

`#(config proxy-client acceleration cifs)` **`directory-cache-time`** *seconds*
Number of seconds for directory listings to remain in the cache. Default is 30.

`#(config proxy-client acceleration cifs)` **{`disable` | `enable`}**
Disable or enable CIFS acceleration. CIFS acceleration is enabled by default.

`#(config proxy-client acceleration cifs)` **`exit`**
Exit the `proxy-client cifs` command.

`#(config proxy-client acceleration cifs)` **`remote-storage-optimization {disable | enable}`**
Enter **`enable`** to cause Windows Explorer to minimize data transfer when users browse to remote accelerated file shares. The amount of performance improvement from enabling ProxyClient remote storage optimization depends on how many files are in the remote folder and how many subfolders are nested under the folder.

Note:

- This feature is not related to Windows offline folders.
- It takes time for a configuration change to take effect. For example, if a client has two tunnels open to an accelerated file share at the time the client receives a configuration update from the Client Manager, it might take several minutes before a change from **`enable`** to **`disable`** takes effect for these open connections.

  On the other hand, the first connection opened to an accelerated file share after a configuration change is received by the client will use the current configuration setting.

`#(config proxy-client acceleration cifs)` **`suppress-folder-customization {disable | enable}`**
Enter **`enable`** to prevent Windows Explorer from displaying folder customizations when users browse to a remote accelerated file share. (An example of folder customization is changing the folder's icon.)

Note:

It takes time for a configuration change to take effect. For example, if a client has two tunnels open to an accelerated file share at the time the client receives a configuration update from the Client Manager, it might take several minutes before a change from **enable** to **disable** takes effect for these open connections.

On the other hand, the first connection opened to an accelerated file share after a configuration change is received by the client will use the current configuration setting.

`#(config proxy-client acceleration cifs)` **`write-back {full | none}`**

Determines whether or not users can continue sending data to the appliance while the appliance is writing data on the back end.

- `full` enables write-back, which in turn makes the appliance appear to the user as a file server; in other words, the appliance constantly sends approval to the client and allows the client to send data while the back end takes advantage of the compressed TCP connection.
- `none` disables write-back. Disabling write-back can introduce substantial latency as clients send data to the appliance and wait for acknowledgement before sending more data.

One reason to set this option to none is the risk of data loss if the link from the branch to the core server fails. There is no way to recover queued data if such a link failure occurs.

`#(config proxy-client acceleration cifs)` **`view`**

View client CIFS settings.

For More Information

- *ProxyClient Administration and Deployment Guide*

Example

```
SGOS#(config proxy-client acceleration cifs) enable
SGOS#(config proxy-client acceleration cifs) write-back full
```

#(config proxy-client locations)

Synopsis

Configure ProxyClient location settings.

Syntax

`#(config) sg-client`

or

`#(config) proxy-client`

This changes the prompt to:

`#(config proxy-client)`

`#(config proxy-client) locations`

This changes the prompt to:

`#(config proxy-client locations)`

Subcommands

`#(config proxy-client locations) acceleration {enable | disable}`
Enable or disable acceleration as a default action; that is, if a client does not match any defined locations.

`#(config proxy-client locations) webfilter {enable | disable}`
Enable or disable Web filtering as a default action; that is, if a client does not match any defined locations.

`#(config proxy-client locations) clear`
Remove all defined locations.

`#(config proxy-client locations) exit`
Exit `locations` submode and return to `proxy-client` mode.

`#(config proxy-client locations) create` *name*
Create location *name*.

`#(config proxy-client locations) delete` *name*
Delete location *name*.

`#(config proxy-client locations) view`
View proxy client location settings.

`#(config proxy-client locations) edit` *name*
Edit location *name*. Changes to the `#(config proxy-client` *name*`)` mode.

`#(config proxy-client` *name*`) exit`

Exit edit submode and return to `proxy-client` mode.

`#(config proxy-client` *name*`) dns`

Define DNS server IP addresses as a condition for this location.

`#(config proxy-client` *name* `dns) add` *ip-address*

Add a DNS server IP address as a location condition. DNS servers are logically ANDed together so a user must match *all* DNS servers defined to match this condition.

`#(config proxy-client` *name* `dns) clear`

Clear all DNS server IP addresses.

`#(config proxy-client` *`name`* `dns)` **`exit`**

Exit the `dns` submode and return to the `proxy-client` *`name`* mode.

`#(config proxy-client` *`name`* `dns)` **`remove`** *`ip-address`*

Remove a DNS server IP address from the location condition.

`#(config proxy-client` *`name`* `dns)` **`view`**

View the list of DNS servers in this location.

`#(config proxy-client` *`name`*`)` **`source`**

Define source IP addresses as a condition for this location.

`#(config proxy-client` *`name`* `source)` **`add`** *`ip-address-range`*

Add a source IP address range as a location condition. Source IP address ranges servers are logically ORd together so a user must log in from any source IP address in any range defined to match this condition.

Source IP address range example: `10.0.0.0-10.255.255.255`

`#(config proxy-client` *`name`* `source)` **`clear`**

Clear all IP source address ranges.

`#(config proxy-client` *`name`* `source)` **`exit`**

Exit the `source` submode and return to the `proxy-client` *`name`* mode.

`#(config proxy-client` *`name`* `source)` **`remove`** *`ip-address-range`*

Remove a source IP address range from the location condition.

Source IP address range example: `10.0.0.0-10.255.255.255`

`#(config proxy-client` *`name`* `source)` **`view`**

View the list of IP source address ranges in this location.

`#(config proxy-client` *`name`*`)` **`vnic`**

Define virtual NIC IP addresses as a condition for this location. Virtual NIC IP address ranges should be used for clients that log in using VPN software that creates a virtual network adapter (also referred to as a *virtual NIC*) that is assigned its own IP address.

`#(config proxy-client` *`name`* `vnic)` **`add`** *`vnic-address-range`*

Add a VNIC IP address range as a location condition. VNIC IP address ranges servers are logically ORd together so a user must log in from any VNIC IP address in any range defined to match this condition.

VNIC IP address range example: `10.0.0.0-10.255.255.255`

`#(config proxy-client` *`name`* `vnic)` **`clear`**

Clear all VNIC IP address ranges.

`#(config proxy-client` *`name`* `vnic)` **`exit`**

Exit the `vnic` submode and return to the `proxy-client` *`name`* mode.

`#(config proxy-client` *`name`* `vnic)` **`remove`** *`vnic-address-range`*

Remove a VNIC IP address range from the location condition.

VNIC IP address range example: `10.0.0.0-10.255.255.255`

`#(config proxy-client` *`name`* `vnic)` **`view`**

View the list of VNIC IP address ranges in this location.

`#(config proxy-client` *`name`*`)` **`match-dns {enable | disable}`**

Enable or disable the use of DNS server IP address as a location condition.

`#(config proxy-client` *name*`)` **`match-source {enable | disable}`**

Enable or disable the use of source IP address ranges as a location condition.

`#(config proxy-client` *name*`)` **`match-vnic {enable | disable}`**

Enable or disable the use of VNIC IP address ranges as a location condition.

`#(config proxy-client` *name*`)` **`acceleration {enable | disable}`**

Enable or disable acceleration for this location. This setting enables or disables all forms of acceleration (that is, gzip, CIFS protocol optimization, and byte caching).

`#(config proxy-client` *name*`)` **`webfilter {enable | disable}`**

Enable or disable Web filtering for this location.

`#(config proxy-client locations)` **`{promote`** *`location-name`* **`| demote`** *`location-name`*`}`

Moves the specified *`location-name`* up or down in the location rulebase. When a ProxyClient connects to the Client Manager, the first match is applied. You should order locations in the rulebase from most specific to least specific. For example, put a location with a source address range from 10.3.0.0 to 10.3.255.255 before a location with the 10.3.0.0. to 10.3.255.255 source address range first.

Use the `#(config proxy-client locations) view` command to view the current location rulebase.

For More Information

- *ProxyClient Administration and Deployment Guide*

Example

The following example creates a location named Mobile, adds two location conditions to it (DNS server IP address and source IP address range), and enables acceleration and Web filtering for the location.

```
#(config proxy-client locations) create Mobile
#(config proxy-client locations) edit Mobile
#(config proxy-client Mobile) dns
#(config proxy-client Mobile dns) add 198.162.1.10
#(config proxy-client Mobile dns) exit
#(config proxy-client Mobile) match-dns enable

#(config proxy-client Mobile) source
#(config proxy-client Mobile source) add 198.162.0.0-198.162.0.255
#(config proxy-client Mobile source) exit
#(config proxy-client Mobile) match-source enable

#(config proxy-client Mobile) acceleration enable
#(config proxy-client Mobile) webfilter enable
```

#(config proxy-client web-filtering)

Synopsis

Configure ProxyClient Web filtering settings.

Syntax

```
#(config) sg-client
```

or

```
#(config) proxy-client
```

This changes the prompt to:

```
#(config proxy-client)
```

```
#(config proxy-client) web-filtering
```

This changes the prompt to:

```
#(config proxy-client web-filtering)
```

Subcommands

`#(config proxy-client web-filtering) {enable | disable}`
Enable or disable ProxyClient Web filtering.

Note: Before you can enable ProxyClient Web filtering, you must obtain a valid Blue Coat WebFilter license. If the Client Manager also performs Web filtering for in-office users, you must enable the Blue Coat Web Filter database on the Client Manager. For more information, see **#(config) content-filter** on page 154.

`#(config proxy-client web-filtering) {allow | block | warn} category_name`
Sets the default action to allow, block, or warn users and groups if they try to access content in this category. Before you can allow, block, or warn users or groups individually using the **user-group-rules** *category_name* command, you must use this command to set the default action for the category.

Content can be from any of the following sources:

- BCWF database categories
- Local database categories; for more information, see **#(config local)** on page 166
- Policy categories; for more information, see *Volume 10: Content Policy Language Guide*.
- System category or Default Action

`#(config proxy-client web-filtering) user-group-rules category_name`
Sets up rules for users and user groups for a category.

`#(config proxy-client web-filtering category) {allow | block | warn} user_group_name`
Allows, blocks, or warns users and groups accessing content in this category. Before you can use this command, you must set the default action for the category. If the *user_group_name* you enter does not already exist, it is created. User and group names can be in any of the following formats:

- Fully qualified account names (for example, *domain_name**user_name*). You should avoid using isolated names (for example, *user_name*).
- Fully qualified DNS names (for example, example.example.com*user_name*)
- User principal names (UPN) (for example, someone@example.com).

`#(config proxy-client web-filtering` *`category`*`) {`**`promote`**` | `**`demote`**`} `*`user_group_name`*
Moves *`user_group_name`* up or down one position in *`category`* the Web filtering rulebase. Policy actions (allow, block, warn) are applied to the first rule that matches the URL request. An error displays if you attempt to promote a user-group rule that is already first in the category or if you attempt to demote a user-group rule that is already last in the category.

`#(config proxy-client web-filtering` *`category`*`)` **`view`**
For the selected category, displays the default action and all user-group rules.

`#(config proxy-client web-filtering` *`category`*`) {`**`promote-to-top`**` | `**`demote-to-bottom`**`} `*`user_group_name`*
Moves *`user_group_name`* to the top or bottom in *`category`* in the Web filtering rulebase. An error displays if you attempt to promote a category that is already first in the rulebase or if you attempt to demote a category that is already last in the rulebase.

`#(config proxy-client web-filtering` *`category`*`)` **`clear`** *`user_group_name`*
Removes *`user_group_name`* from this category.

`#(config proxy-client web-filtering)` **`clear`** `{`**`all`**` | `*`category_name`*`}`
Clears all or the specified category from the rulebase.

`#(config proxy-client web-filtering)` **`default-action`** `{`**`allow`**` | `**`block`**`}`
Set the default action to take in the event the user requests content that is not classified in any category you selected.

`#(config proxy-client web-filtering) {`**`promote`**` | `**`demote`**`}` *`category_name`*
Moves *`category_name`* (including all users and groups defined for this category) up or down one position in the Web filtering rulebase. Policy actions (allow, block, warn) are applied to the first rule that matches the URL request. Because URLs are typically classified in more than one category, the rulebase order is important. An error displays if you attempt to promote a category that is already first in the rulebase or if you attempt to demote a category that is already last in the rulebase.

`#(config proxy-client web-filtering) {`**`promote-to-top`**` | `**`demote-to-bottom`**`}` *`category_name`*
Moves *`category_name`* (including all users and groups defined for this category) to the top or bottom of the Web filtering rulebase. An error displays if you attempt to promote a category that is already first in the rulebase or if you attempt to demote a category that is already last in the rulebase.

`#(config proxy-client web-filtering)` **`exit`**
Exit `web-filtering` submode and return to `proxy-client` mode.

`#(config proxy-client web-filtering)` **`failure-mode`** `{`**`closed`**` | `**`open`**`}`
Specify the action to take if the BCWF license expires (usually because the database has not been updated in a 30-day period). **`closed`** means users are not allowed to browse to any Web page. A Service Unavailable exception displays in the user's Web browser. **`open`** means users are allowed to browse anywhere; in other words, content is not filtered. Select this option if user Web access is more critical than filtering or security.

`#(config proxy-client web-filtering)` **`https-filtering`** `{`**`disable`**` | `**`enable`**`}`
Set to **`enable`** to use Web filtering when the content request is sent over an SSL connection using the default port 443. For exceptions to this behavior, see the *ProxyClient Release Notes*. Set to **`disable`** to not filter HTTPS traffic from unsupported browsers.

`#(config proxy-client web-filtering)` **`safe-search`** `{`**`disable`**` | `**`enable`**`}`
Set to **`enable`** to force a search engine that supports Safe Search to enable its strictest search filter; however, the quality of the filtering is based on the given engine's built-in capabilities. The same search string entered on one search engine might yield different results when entered on another search engine (including varying levels of inappropriate content). Safe Search is supported on the following search engines: Google, A9, Altavista, Microsoft Live, Yahoo, Ask, and Orange.co.uk. With safe search enabled, the search engine Web page displays Safe Search ON, Family Filter On, Safe Search Strict, or another engine-specific string. Set to **`disable`** if you do not wish to enforce Safe Search.

`#(config proxy-client web-filtering)` **`inline exception {block | warn | unavailable}`** *`data end-of-file-marker`*

Sets up exception pages to display to users when they attempt to access certain content. Set the exception page for **`block`** to display a page when users attempt to access blocked content. Set the exception page for **`warn`** to display a page when users attempt to access content that might violate company policies. Set the exception page for **`unavailable`** to display a page when users attempt to access content that cannot be categorized because the service point is not available.

`data` is the HTML code to display to users.

`end-of-file-marker` is discussed in the section on Tips in **`#(config) inline`** on page 227.

`#(config proxy-client web-filtering)` **`log`**

`#(config proxy-client log)` **`{disable | enable}`**
Enable or disable uploading of ProxyClient Web filtering user logs to an anonymous FTP server.

`#(config proxy-client log)` **`exit`**
Exit `log` submode and return to `proxy-client` mode.

`#(config proxy-client log)` **`ftp-client {{primary | alternate} {host`** *`host-or-ip-address`* **`[port`** *`port`*`]} {`**`path`** *`path`*`}}`

Specify the anonymous FTP server to which users upload ProxyClient Web filtering logs and the *`path`* to which to upload the files. You can optionally precede the relative path with the **`/`** character; uploads will succeed whether or not the first character is **`/`**.

Examples:

`/path/to/log/directory`

`path/to/log/directory`

To upload logs to the FTP server's home directory, leave the field blank.

Note: Entering **`/`** in the field (with no path following the **`/`** character) causes uploads to fail.

Note: Because log files are uploaded using anonymous FTP, Blue Coat strongly recommends you put your FTP server behind the corporate firewall. In addition, you should configure the FTP server as follows:

- To prevent the possibility of data loss, do not allow file overwrites.
- For security reasons, do not allow files on the FTP server's upload directory to be browsed.
- The FTP server must support passive FTP clients.
- If the FTP server is deployed behind a firewall, the firewall must be configured to allow FTP data connections over TCP ports greater than 1024.
- Placing an FTP server outside the firewall has the advantage that even mobile users can upload log files to it; however, it exposes the server and your company to potentially serious malicious activity

---

`#(config proxy-client log)` **`mode {all-requests | exceptions-only}`**
Enter **`all-requests`** to upload the entire client log. Enter **`exceptions-only`** to upload only exceptions.

`#(config proxy-client log)` **`periodic-upload upload-interval`** *`{hours [minutes]}`*
Enter the number of hours for clients to attempt to upload their logs to the anonymous FTP server. Optionally enter the number of minutes, in addition to *`hours`*.
Note: If you enter a non-zero value for both *`hours`* and *`minutes`*, the total amount of time is used. For example, if you enter **`periodic-upload 24 10`**, the client waits 24 hours and 10 minutes to upload log files.

A change to the upload period does not take effect immediately. In other words, if the upload period is 24 hours and you change it to 20 hours, clients with the 24 hour configuration wait 24 hours to upload their current logs before the 20 hour upload period takes effect.

`#(config proxy-client log)` **`early-upload`** *`megabytes`*
Enter the maximum log file size, in megabytes, to trigger a log file upload. This value takes precedence over the **`periodic-upload`** parameter. In other words, if you specify **`periodic-upload 24`** and **`early-upload 10`**, if the client log file size reaches 10 megabytes after only 10 hours, the ProxyClient attempts to upload its log files to the FTP server.

`#(config proxy-client` log) **`view`**
View current ProxyClient Web filtering log settings.

`#(config proxy-client web-filtering)` **`view`**
View current ProxyClient Web filtering settings.

For More Information

- *ProxyClient Administration and Deployment Guide*

Example

The following example enables Web filtering, sets up two categories—Sports/Recreation (set to deny) and News/Media (set to allow)—and sets other options.

```
#(config proxy-client web-filtering) enable
#(config proxy-client web-filtering) allow News/Media
#(config proxy-client web-filtering) block Sports/Recreation
#(config proxy-client web-filtering) default-action allow
#(config proxy-client web-filtering) https-filtering enable
#(config proxy-client web-filtering) failure-mode closed
```

The following example enables Web filtering, sets up two categories—Sports/Recreation (blocked for everyone in the `BLUECOAT\Engineering` group) and News/Media (allowed for the user `raymond.marcom@example.com`)—and sets other options.

```
#(config proxy-client web-filtering) enable
#(config proxy-client web-filtering) block News/Media
#(config proxy-client web-filtering) user-group-rules News/Media
#(config proxy-client web-filtering News/Media) allow raymond.marcom@example.com
#(config proxy-client web-filtering) allow Sports/Recreation
#(config proxy-client web-filtering) user-group-rules Sports/Recreation
#(config proxy-client web-filtering Sports/Recreation) block
BLUECOAT\\Engineering
```

#(config) proxy-services

Synopsis

Manages the proxy services on the ProxySG.

Syntax

```
#(config) proxy-services
```

This changes the prompt to:

```
#(config proxy-services)
```

Subcommands

Note: Additional information is found under options that are hyperlinked (blue).

`#(config proxy-services)` **`create`** *`service_type service_name`* `[`*`service_group`*`]`
Creates a proxy service of the type and name that you specify. Optionally, specify a service group. If no service group is specified the service is placed in the service group "Other."For more information on creating specific proxy services, see "Available Service Types" on page 276.

`#(config proxy-services)` **`delete`** *`service_name`*
Deletes the specified proxy service.

`#(config proxy-services)` **`dynamic-bypass`**
Changes the prompt to **`#(config dynamic-bypass)`** on page 282 to allow you to manage dynamic-bypass settings.

`#(config proxy-services)` **`edit`** *`service_name`*
Allows you to edit a proxy service of the specified name. For more information on editing specific proxy services, see "Available Service Types" on page 276.

`#(config proxy-services)` **`exit`**
Returns to the `#(config)` prompt.

`#(config proxy-services)` **`force-bypass`** {**`disable`** | **`enable`**}
Allows you to temporarily bypass all proxy services when enabled. Disabling force bypass returns proxy services to normal operation.

`#(config proxy-services)` **`import`** {*`predefined-service`* | **`overwrite`**}
Imports a predefined service from the library. Optionally, an existing service may be replace by a service from the library by entering the keyword **`overwrite`**.

`#(config proxy-services)` **`restricted-intercept`**
Changes the prompt to **`#(config restricted-intercept)`** on page 294 to allow you to restrict interception to a limited number of clients and servers.

`#(config proxy-services)` **`static-bypass`**
Changes the prompt to **`#(config static-bypass)`** on page 300 to allow you to manage static-bypass settings.

`#(config proxy-services)` **`view`** {**`dynamic-bypass`** | **`services`** | **`static-bypass`**}
Allows you to view proxy service parameters.

Available Service Types

You can create proxy services using the following proxies:

Note: The service types listed below are not necessarily the service names you use. The syntax for creating a service type is `#(config proxy-services) create service_type service_name`, where *service_type* is one of those listed below and *service_name* is of your choosing.

- **#(config aol-im)** on page 277
- **#(config cifs)** on page 278
- **#(config dns)** on page 280
- **#(config Endpoint Mapper)** on page 284
- **#(config ftp)** on page 286
- **#(config HTTP)** on page 288
- **#(config https-reverse-proxy)** on page 290
- **#(config mms)** on page 292
- **#(config msn-im)** on page 293
- **#(config rtsp)** on page 296
- **#(config socks)** on page 297
- **#(config ssl)** on page 298
- **#(config tcp-tunnel)** on page 301
- **#(config telnet)** on page 303
- **#(config yahoo-im)** on page 305

For More Information

- *SGOS Administration Guide*

Example

```
#(config proxy-services) create tcp-tunnel tcp_tunnel_2
  ok
#(config proxy-services) edit tcp_tunnel_2
#(config tcp_tunnel_2)?
add                         Add a listener
attribute                   Configure service attributes
bypass                      Change a particular listener's action to bypass
exit                        Return to (config proxy-services) prompt
intercept                   Change a particular listener's action to intercept
remove                      Remove a listener
view                        Show proxy service configuration
```

#(config aol-im)

Synopsis

Allows you to manage services that are controlled by the AOL-IM proxy.

Syntax

```
#(config proxy-services) create service_type service_name
#(config proxy-services) edit service_name
```

This changes the prompt to:

`#(config service_name)`

Subcommands

`#(config service_name) add {all | source_ip | source_ip/subnet-mask} {transparent | explicit | all | destination_ip | destination_ip/subnet-mask} {port | first_port-last_port} [intercept | bypass]`
Allows you to add a listener with the parameters you specify.

`#(config service_name) bypass {all | source_ip | source_ip/subnet-mask} {transparent | explicit | all | destination_ip | destination_ip/subnet-mask} {port | first_port-last_port}`
Changes the behavior from intercept to bypass for the listener you specify.

`#(config service_name) exit`
Exits to the `#(config proxy-services)` prompt.

`#(config service_name) group service-group`

Allows you to move a particular service to another service group.

`#(config service_name) intercept {{all | source_ip | source_ip/subnet-mask} {transparent | explicit | all | destination_ip | destination_ip/subnet-mask} {port | first_port-last_port}`
Changes the behavior from bypass to intercept for the listener you specify.

`#(config service_name) proxy-type proxy-type`
Allows you to change the proxy type of a particular service.

`#(config service_name) remove {all | source_ip | source_ip/subnet-mask} {transparent | explicit | all | destination_ip | destination_ip/subnet-mask} {port | first_port-last_port}`
Allows you to remove a listener with the parameters you specify.

`#(config service_name) view`
Views the specified proxy service.

For More Information

- *SGOS Administration Guide*

Example

```
SGOS#(config proxy-services) create aol-im aol1
SGOS#(config proxy-services) edit aol1
```

#(config cifs)

Synopsis

Allows you to manage services that are controlled by the CIFS proxy.

Syntax

```
#(config proxy-services) create service_type service_name
#(config proxy-services) edit service_name
```

This changes the prompt to:

```
#(config service_name)
```

Subcommands

`#(config service_name) add {all | source_ip | source_ip/subnet-mask} {transparent | destination_ip | destination_ip/subnet-mask} {port | first_port-last_port} [intercept|bypass]`
Allows you to add a listener with the parameters you specify.

`#(config service_name) attribute use-adn {disable | enable}`
Controls whether ADN is enabled for a specific service. Enabling ADN does not guarantee the connections are accelerated by ADN. The actual decision is determined by ADN routing (for explicit deployment) and network setup (for transparent deployment).

`#(config service_name) attribute adn-byte-cache {disable | enable}`
Controls whether to optimize traffic using the byte caching optimization technique when connecting upstream in an ADN tunnel.

`#(config service_name) attribute adn-compress {disable | enable}`
Controls whether to optimize traffic using GZIP compression when connecting upstream in an ADN tunnel.

`#(config service_name) attribute adn-optimize {disable | enable}`
Starting in SGOS 6.2, the `adn-optimize` command was replaced by the `adn-byte-cache` and `adn-compress` commands (see above).

`#(config service_name) attribute byte-cache-priority {low | normal | high}`
Adjust retention priority of byte cache data. If you want to keep certain types of streams in the byte cache for as long as possible, set a high retention priority for the service. Or for streams that aren't likely to get much benefit from byte caching, you can set a low retention priority for the related service. Most services are set to normal priority by default. Note that unless the underlying service has `adn-byte-cache` enabled, the priority setting will have no effect; if you try to set a retention priority when byte caching is disabled, a warning message displays to inform you that the `byte-cache-priority` attribute has no effect when `adn-byte-cache` is disabled.

`#(config service_name) bypass {all | source_ip | source_ip/subnet-mask} {transparent | destination_ip | destination_ip/subnet-mask} {port | first_port-last_port}`
Change the behavior from intercept to bypass for the listener you specify.

`#(config service_name) exit`
Exits to the `#(config proxy-services)` prompt.

`#(config service_name) group service-group`
Allows you to move a particular service to another service group.

`#(config service_name) intercept {all | source_ip | source_ip/subnet-mask} {transparent | destination_ip | destination_ip/subnet-mask} {port | first_port-last_port}`
Change the behavior from bypass to intercept for the listener you specify.

`#(config service_name) proxy-type proxy-type`
Allows you to change the proxy type of a particular service.

`#(config service_name) remove {all | source_ip | source_ip/subnet-mask} {transparent | destination_ip | destination_ip/subnet-mask} {port | first_port-last_port}`
Allows you to remove a listener with the parameters you specify.

`#(config service_name) view`
Views the specified proxy service.

For More Information

- *SGOS Administration Guide*

Example

```
SGOS#(config proxy-services) create cifs cifs1
SGOS#(config proxy-services) edit cifs1
SGOS #(config cifs1) attribute adn-byte-cache disable
ok
```

#(config dns)

Synopsis

Allows you to manage services that are controlled by the DNS proxy.

Syntax

```
#(config proxy-services) create service_type service_name
#(config proxy-services) edit service_name
```

This changes the prompt to:

```
#(config service_name)
```

Subcommands

`#(config service_name) add {all | source_ip | source_ip/subnet-mask} {transparent | explicit | all | destination_ip | destination_ip/subnet-mask} {port | first_port-last_port} [intercept | bypass]`
Allows you to add a listener with the parameters you specify. IP addresses can be IPv4 or IPv6; enter a subnet mask (for IPv4) or prefix length (for IPv6). When the destination address on a listener is defined as **transparent**, **explicit**, or **all**, it applies to IPv4 and IPv6 addresses.

`#(config service_name) bypass {all | source_ip | source_ip/subnet-mask} {transparent | explicit | all | destination_ip | destination_ip/subnet-mask} {port | first_port-last_port}`
Change the behavior from intercept to bypass for the listener you specify. IP addresses can be IPv4 or IPv6; enter a subnet mask (for IPv4) or prefix length (for IPv6). When the destination address on a listener is defined as **transparent**, **explicit**, or **all**, it applies to IPv4 and IPv6 addresses.

`#(config service_name) exit`
Exits to the `# (config proxy-services)` prompt.

`#(config service_name) group service-group`
Allows you to move a particular service to another service group.

`#(config service_name) intercept {all | source_ip | source_ip/subnet-mask} {transparent | explicit | all | destination_ip | destination_ip/subnet-mask} {port | first_port-last_port}}`
Change the behavior from bypass to intercept for the listener you specify. IP addresses can be IPv4 or IPv6; enter a subnet mask (for IPv4) or prefix length (for IPv6). When the destination address on a listener is defined as **transparent**, **explicit**, or **all**, it applies to IPv4 and IPv6 addresses.

`#(config service_name) proxy-type proxy-type`
Allows you to change the proxy type of a particular service.

`#(config service_name) remove {all | source_ip | source_ip/subnet-mask} {transparent | explicit | all | destination_ip | destination_ip/subnet-mask} {port | first_port-last_port} [intercept | bypass]`
Allows you to remove a listener with the parameters you specify. IP addresses can be IPv4 or IPv6; enter a subnet mask (for IPv4) or prefix length (for IPv6).

`#(config service_name) view`
Views the specified proxy service.

For More Information

- *SGOS Administration Guide*

Example

```
SGOS#(config proxy-services) create dns dns1
SGOS#(config proxy-services) edit dns1
```

#(config dynamic-bypass)

Synopsis

Dynamic bypass provides a maintenance-free method for improving performance of the ProxySG by automatically compiling a list of requested URLs that return various kinds of errors.

Syntax

```
#(config) proxy-services
#(config proxy-services) dynamic-bypass
```

The prompt changes to:

```
#(config dynamic-bypass)
```

Subcommands

`#(config dynamic-bypass) clear`
Clears all dynamic bypass entries.

`#(config dynamic-bypass) disable`
Disables dynamic bypass.

`#(config dynamic-bypass) enable`
Enables dynamic bypass.

`#(config dynamic-bypass) exit`
Exits to the `#(config proxy-services)` prompt.

`#(config dynamic-bypass) max-entries` *number_of_entries*
Specifies the maximum number of dynamic-bypass entries. Connections that match entries in the dynamic bypass list are not intercepted by the application proxies. Entries in the dynamic bypass list eventually time out based on the configuration. If the list grows beyond its configured size, the oldest entry is removed

`#(config dynamic-bypass) no trigger {all | connect-error | non-http | receive-error | 400 | 401 | 403 | 405 | 406 | 500 | 502 | 503 | 504}`
Disables dynamic bypass for the specified HTTP response code, all HTTP response codes, or all non-HTTP responses. Values are specified below.

Event Value	Description
all	Enables all dynamic bypass triggers.
non-http	Enables dynamic bypass for non-HTTP responses.
connect-error	Enables dynamic bypass for any connection failure to the origin content server, including timeouts.
receive-error	Enables dynamic bypass for when a TCP connection to an origin content server succeeds, but the cache does not receive an HTTP response.
400	Enables dynamic bypass for HTTP 400 responses.
401	Enables dynamic bypass for HTTP 401 responses.
403	Enables dynamic bypass for HTTP 403 responses.
405	Enables dynamic bypass for HTTP 405 responses.
406	Enables dynamic bypass for HTTP 406 responses.

Event Value	Description
500	Enables dynamic bypass for HTTP 500 responses.
502	Enables dynamic bypass for HTTP 502 responses.
503	Enables dynamic bypass for HTTP 503 responses.
504	Enables dynamic bypass for HTTP 504 responses.

`#(config dynamic-bypass) server-threshold` *`number_of_entries`*
Specifies the number of client entries for all clients to bypass a server. Each dynamic entry can be identified by a server address or client/server address pair. A dynamic entry without a client address means the client address is a wildcard address. For example, if the server threshold is set to 10 and there are already nine dynamic entries with different client addresses for the same server address, the next time a new dynamic entry is added to the same server address but contains a different client address, the ProxySG compresses the nine dynamic entries into one dynamic entry with server address only; all clients going to that server address are bypassed.

`#(config dynamic-bypass) timeout` *`minutes`*
Sets the dynamic-bypass timeout interval in minutes.

`#(config dynamic-bypass) trigger {all | connect-error | non-http | receive-error | 400 | 401 | 403 | 405 | 406 | 500 | 502 | 503 | 504}`
Enables dynamic bypass for the specified HTTP response code, all HTTP response codes, or all non-HTTP responses.

`#(config dynamic-bypass) view {configuration | filter {* | all |` *`client_ip_address | client_ip_address/subnet-mask`* `} {* | all |` *`server_ip_address | server_ip_address/subnet-mask`* `}} | <Enter>}`
Allows you to view the dynamic-bypass configuration or to filter the dynamic-bypass list on the parameters above.

For More Information

- *SGOS Administration Guide*

Example

```
#(config) proxy-services
#(config proxy-services) dynamic-bypass
#(config dynamic-bypass) clear
 ok
#(config dynamic-bypass) enable
WARNING:
       Requests to sites that are put into the dynamic bypass list will
       bypass future policy evaluation. This could result in subversion
       of on-box policy. The use of dynamic bypass is cautioned.
 ok
#(config dynamic-bypass) trigger all
 ok
```

#(config Endpoint Mapper)

Synopsis

Allows you to manage services that are controlled by the Endpoint Mapper proxy.

Syntax

```
#(config proxy-services) create service_type service_name
#(config proxy-services) edit service_name
```

This changes the prompt to:

```
#(config service_name)
```

Subcommands

`#(config proxy-services service_name) add {all | source_ip | source_ip/subnet-mask} {destination_ip | destination_ip/subnet-mask} {port | first_port-last_port} [intercept | bypass]`
Allows you to add a listener with the parameters you specify.

`#(config service_name) attribute use-adn {disable | enable}`
Controls whether ADN is enabled for a specific service. Enabling ADN does not guarantee the connections are accelerated by ADN. The actual decision is determined by ADN routing (for explicit deployment) and network setup (for transparent deployment).

`#(config service_name) attribute adn-byte-cache {disable | enable}`
Controls whether to optimize traffic using the byte caching optimization technique when connecting upstream in an ADN tunnel.

`#(config service_name) attribute adn-compress {disable | enable}`
Controls whether to optimize traffic using GZIP compression when connecting upstream in an ADN tunnel.

`#(config service_name) attribute adn-optimize {disable | enable}`
Starting in SGOS 6.2, the `adn-optimize` command was replaced by the `adn-byte-cache` and `adn-compress` commands (see above).

`#(config service_name) attribute byte-cache-priority {low | normal | high}`
Adjust retention priority of byte cache data. If you want to keep certain types of streams in the byte cache for as long as possible, set a high retention priority for the service. Or for streams that aren't likely to get much benefit from byte caching, you can set a low retention priority for the related service. Most services are set to normal priority by default. Note that unless the underlying service has `adn-byte-cache` enabled, the priority setting will have no effect; if you try to set a retention priority when byte caching is disabled, a warning message displays to inform you that the `byte-cache-priority` attribute has no effect when `adn-byte-cache` is disabled.

`#(config service_name) bypass {all | source_ip | source_ip/subnet-mask} {destination_ip | destination_ip/subnet-mask} {port | first_port-last_port}`
Change the behavior from intercept to bypass for the listener you specify.

`#(config service_name) exit`
Exits to the `#(config proxy-services)` prompt.

`#(config service_name) group service-group`
Allows you to move a particular service to another service group.

`#(config service_name) intercept {all | source_ip | source_ip/subnet-mask} {destination_ip | destination_ip/subnet-mask} {port | first_port-last_port}`
Change the behavior from bypass to intercept for the listener you specify.

`#(config service_name) proxy-type proxy-type`
Allows you to change the proxy type of a particular service.

`#(config service_name) remove {all | source_ip | source_ip/subnet-mask} {destination_ip | destination_ip/subnet-mask} {port | first_port-last_port} [intercept | bypass]`
Allows you to remove a listener with the parameters you specify.

`#(config service_name) view`
Views the specified proxy service.

For More Information

- *SGOS Administration Guide*

Example

```
SGOS#(config proxy-services) create endpoint-mapper epmapper1
SGOS#(config proxy-services) edit epmapper1
SGOS#(config epmapper1) add all 10003
ok
```

#(config ftp)

Synopsis

Allows you to manage services that are controlled by the FTP proxy.

Syntax

```
#(config proxy-services) create service_type service_name
#(config proxy-services) edit service_name
```

This changes the prompt to:

```
#(config service_name)
```

Subcommands

`#(config service_name) add {all | source_ip | source_ip/subnet-mask} {transparent | explicit | all | destination_ip | destination_ip/subnet-mask} {port | first_port-last_port} [intercept | bypass]`
Allows you to add a listener with the parameters you specify. IP addresses can be IPv4 or IPv6; enter a subnet mask (for IPv4) or prefix length (for IPv6). When the destination address on a listener is defined as **transparent**, **explicit**, or **all**, it applies to IPv4 and IPv6 addresses.

`#(config service_name) attribute use-adn {disable | enable}`
Controls whether ADN is enabled for a specific service. Enabling ADN does not guarantee the connections are accelerated by ADN. The actual decision is determined by ADN routing (for explicit deployment) and network setup (for transparent deployment).

`#(config service_name) attribute adn-byte-cache {disable | enable}`
Controls whether to optimize traffic using the byte caching optimization technique when connecting upstream in an ADN tunnel.

`#(config service_name) attribute adn-compress {disable | enable}`
Controls whether to optimize traffic using GZIP compression when connecting upstream in an ADN tunnel.

`#(config service_name) attribute adn-optimize {disable | enable}`
Starting in SGOS 6.2, the `adn-optimize` command was replaced by the `adn-byte-cache` and `adn-compress` commands (see above).

`#(config service_name) attribute byte-cache-priority {low | normal | high}`
Adjust retention priority of byte cache data. If you want to keep certain types of streams in the byte cache for as long as possible, set a high retention priority for the service. Or for streams that aren't likely to get much benefit from byte caching, you can set a low retention priority for the related service. Most services are set to normal priority by default. Note that unless the underlying service has `adn-byte-cache` enabled, the priority setting will have no effect; if you try to set a retention priority when byte caching is disabled, a warning message displays to inform you that the `byte-cache-priority` attribute has no effect when `adn-byte-cache` is disabled.

`#(config service_name) bypass {all | source_ip | source_ip/subnet-mask} {transparent | explicit | all | destination_ip | destination_ip/subnet-mask} {port | first_port-last_port}`
Change the behavior from intercept to bypass for the listener you specify. IP addresses can be IPv4 or IPv6; enter a subnet mask (for IPv4) or prefix length (for IPv6). When the destination address on a listener is defined as **transparent**, **explicit**, or **all**, it applies to IPv4 and IPv6 addresses.

`#(config service_name) exit`
Exits to the `#(config proxy-services)` prompt.

`#(config service_name) group service-group`
Allows you to move a particular service to another service group.

`#(config service_name)` **`intercept`** `{`**`all |`** *`source_ip`* `|` *`source_ip/subnet-mask`*`}`
`{transparent | explicit | all |` *`destination_ip`* `|` *`destination_ip/subnet-mask`*`}`
`{`*`port`* `|` *`first_port-last_port`*`}`
Change the behavior from bypass to intercept for the listener you specify. IP addresses can be IPv4 or IPv6; enter a subnet mask (for IPv4) or prefix length (for IPv6). When the destination address on a listener is defined as **`transparent`**, **`explicit`**, or **`all`**, it applies to IPv4 and IPv6 addresses.

`#(config service_name)` **`proxy-type`** *`proxy-type`*
Allows you to change the proxy type of a particular service.

`#(config service_name)` **`remove`** `{`**`all |`** *`source_ip`* `|` *`source_ip/subnet-mask`*`}`
`{transparent | explicit | all |` *`destination_ip`* `|` *`destination_ip/subnet-mask`*`}`
`{`*`port`* `|` *`first_port-last_port`*`}`
Allows you to remove a listener with the parameters you specify. IP addresses can be IPv4 or IPv6; enter a subnet mask (for IPv4) or prefix length (for IPv6).

`#(config service_name)` **`view`**
Views the specified proxy service.

For More Information

- *SGOS Administration Guide*

Example

```
SGOS#(config proxy-services) create ftp ftp1
SGOS#(config proxy-services) edit ftp1
SGOS #(config ftp1) intercept all 10004
ok
```

#(config HTTP)

Synopsis

Allows you to manage services that are controlled by the HTTP proxy.

Syntax

```
#(config proxy-services) create service_type service_name
#(config proxy-services) edit service_name
```

This changes the prompt to:

```
#(config service_name)
```

Subcommands

`#(config service_name) add {all | source_ip | source_ip/subnet-mask} {transparent | explicit | all | destination_ip | destination_ip/subnet-mask} {port | first_port-last_port} [intercept | bypass]`
Allows you to add a listener with the parameters you specify. IP addresses can be IPv4 or IPv6; enter a subnet mask (for IPv4) or prefix length (for IPv6). When the destination address on a listener is defined as **transparent**, **explicit**, or **all**, it applies to IPv4 and IPv6 addresses.

`#(config service_name) attribute adn-byte-cache {disable | enable}`
Controls whether to optimize traffic using the byte caching optimization technique when connecting upstream in an ADN tunnel.

`#(config service_name) attribute adn-compress {disable | enable}`
Controls whether to optimize traffic using GZIP compression when connecting upstream in an ADN tunnel.

`#(config service_name) attribute adn-optimize {disable | enable}`
Starting in SGOS 6.2, the `adn-optimize` command was replaced by the `adn-byte-cache` and `adn-compress` commands (see above).

`#(config service_name) attribute authenticate-401 {disable | enable}`
All transparent and explicit requests received on the port always use transparent authentication (cookie or IP, depending on the configuration). This is especially useful to force transparent proxy authentication in some proxy-chaining scenarios.

`#(config service_name) attribute byte-cache-priority {low | normal | high}`
Adjust retention priority of byte cache data. If you want to keep certain types of streams in the byte cache for as long as possible, set a high retention priority for the service. Or for streams that aren't likely to get much benefit from byte caching, you can set a low retention priority for the related service. Most services are set to normal priority by default. Note that unless the underlying service has `adn-byte-cache` enabled, the priority setting will have no effect; if you try to set a retention priority when byte caching is disabled, a warning message displays to inform you that the `byte-cache-priority` attribute has no effect when `adn-byte-cache` is disabled.

`#(config service_name) attribute connect (disable | enable}`
This command is deprecated. Policy should be used instead. For example:

```
; To block CONNECT destined to ports other then 443
<Proxy>
  url.port=!443 http.method=CONNECT deny
```

`#(config service_name) attribute detect-protocol {disable | enable}`
Protocols that can be detected include: HTTP, P2P (eDonkey, BitTorrent, FastTrack, Gnutella), SSL, and Endpoint Mapper.

`#(config service_name) attribute head (disable | enable}`
This command is deprecated. Policy should be used instead. For example:

```
; To block HEAD methods
<Proxy>
  http.method=HEAD deny
```

`#(config service_name) attribute use-adn {disable | enable}`
Controls whether ADN is enabled for a specific service. Enabling ADN does not guarantee the connections are accelerated by ADN. The actual decision is determined by ADN routing (for explicit deployment) and network setup (for transparent deployment).

`#(config service_name) bypass {all | source_ip | source_ip/subnet-mask} {transparent | explicit | all | destination_ip | destination_ip/subnet-mask} {port | first_port-last_port}`
Change the behavior from intercept to bypass for the listener you specify. IP addresses can be IPv4 or IPv6; enter a subnet mask (for IPv4) or prefix length (for IPv6). When the destination address on a listener is defined as **transparent**, **explicit**, or **all**, it applies to IPv4 and IPv6 addresses.

`#(config service_name) exit`
Exits to the `#(config proxy-services)` prompt.

`#(config service_name) group service-group`
Allows you to move a particular service to another service group.

`#(config service_name) intercept {all | source_ip | source_ip/subnet-mask} {transparent | explicit | all | destination_ip | destination_ip/subnet-mask} {port | first_port-last_port}`
Change the behavior from bypass to intercept for the listener you specify. IP addresses can be IPv4 or IPv6; enter a subnet mask (for IPv4) or prefix length (for IPv6). When the destination address on a listener is defined as **transparent**, **explicit**, or **all**, it applies to IPv4 and IPv6 addresses.

`#(config service_name) proxy-type proxy-type`
Allows you to change the proxy type of a particular service.

`#(config service_name) remove {all | source_ip | source_ip/subnet-mask} {transparent | explicit | all | destination_ip | destination_ip/subnet-mask} {port | first_port-last_port}`
Allows you to remove a listener with the parameters you specify. IP addresses can be IPv4 or IPv6; enter a subnet mask (for IPv4) or prefix length (for IPv6).

`#(config service_name) view`
Views the specified proxy service.

For More Information

- *SGOS Administration Guide*

Example

```
SGOS#(config proxy-services) create http http2
SGOS#(config proxy-services) edit http2
SGOS#(config http2) attribute authenticate-401 enable
ok
```

#(config https-reverse-proxy)

Synopsis

Allows you to manage services that are controlled by the HTTPS reverse proxy.

Syntax

```
#(config proxy-services) create service_type service_name
#(config proxy-services) edit service_name
```

This changes the prompt to:

```
#(config service_name)
```

Subcommands

`#(config service_name) add {all | source_ip | source_ip/subnet-mask} {transparent | explicit | all | destination_ip | destination_ip/subnet-mask} {port | first_port-last_port} [intercept | bypass]`
Allows you to add a listener with the parameters specified. IP addresses can be IPv4 or IPv6; enter a subnet mask (for IPv4) or prefix length (for IPv6). When the destination address on a listener is defined as **transparent**, **explicit**, or **all**, it applies to IPv4 and IPv6 addresses.

`#(config service_name) attribute use-adn {disable | enable}`
Controls whether ADN is enabled for a specific service. Enabling ADN does not guarantee the connections are accelerated by ADN. The actual decision is determined by ADN routing (for explicit deployment) and network setup (for transparent deployment).

`#(config service_name) attribute adn-byte-cache {disable | enable}`
Controls whether to optimize traffic using the byte caching optimization technique when connecting upstream in an ADN tunnel.

`#(config service_name) attribute adn-compress {disable | enable}`
Controls whether to optimize traffic using GZIP compression when connecting upstream in an ADN tunnel.

`#(config service_name) attribute adn-optimize {disable | enable}`
Starting in SGOS 6.2, the `adn-optimize` command was replaced by the `adn-byte-cache` and `adn-compress` commands (see above).

`#(config service_name) attribute byte-cache-priority {low | normal | high}`
Adjust retention priority of byte cache data. If you want to keep certain types of streams in the byte cache for as long as possible, set a high retention priority for the service. Or for streams that aren't likely to get much benefit from byte caching, you can set a low retention priority for the related service. Most services are set to normal priority by default. Note that unless the underlying service has `adn-byte-cache` enabled, the priority setting will have no effect; if you try to set a retention priority when byte caching is disabled, a warning message displays to inform you that the `byte-cache-priority` attribute has no effect when `adn-byte-cache` is disabled.

`#(config service_name) attribute ccl list_name`
CA Certificate List used for verifying client certificates.

`#(config service_name) attribute cipher-suite cipher-suite+`
Allows you to specify the cipher suites you want to use with the https-reverse-proxy service.

`#(config service_name) attribute forward-client-cert {disable | enable}`
When used with the `verify-client` attribute, puts the extracted client certificate information into a header that is included in the request when it is forwarded to the OCS. The name of the header is Client-Cert. The header contains the certificate serial number, subject, validity dates and issuer (all as name=value pairs). The actual certificate is not forwarded.

`#(config service_name) attribute keyring keyring-ID`
Allows you to specify the keyring you want to use with this service.

`#(config service_name) attribute ssl-versions {sslv2 |sslv3 | tlsv1 | tlsv1.1 | tls1.2}`
Allows you to select which versions of SSL you want to support. The default is to enable TLS version 1, 1.1, and 1.2.

`#(config service_name) attribute verify-client {disable | enable}`
Requests and validates the SSL client certificate.

`#(config service_name) bypass {all | source_ip | source_ip/subnet-mask} {transparent | explicit | all | destination_ip | destination_ip/subnet-mask} {port | first_port-last_port}`
Changes the behavior from intercept to bypass for the listener specified. IP addresses can be IPv4 or IPv6; enter a subnet mask (for IPv4) or prefix length (for IPv6). When the destination address on a listener is defined as **transparent**, **explicit**, or **all**, it applies to IPv4 and IPv6 addresses.

`#(config service_name) exit`
Exits to the `#(config proxy-services)` prompt.

`#(config service_name) group service-group`
Allows you to move a particular service to another service group.

`#(config service_name) intercept {all | source_ip | source_ip/subnet-mask} {transparent | explicit | all | destination_ip | destination_ip/subnet-mask} {port | first_port-last_port}`
Change the behavior from bypass to intercept for the listener you specify. IP addresses can be IPv4 or IPv6; enter a subnet mask (for IPv4) or prefix length (for IPv6). When the destination address on a listener is defined as **transparent**, **explicit**, or **all**, it applies to IPv4 and IPv6 addresses.

`#(config service_name) proxy-type proxy-type`
Allows you to change the proxy type of a particular service.

`#(config service_name) remove {all | source_ip | source_ip/subnet-mask} {transparent | explicit | all | destination_ip | destination_ip/subnet-mask} {port | first_port-last_port}`
Allows you to remove a listener with the parameters you specify. IP addresses can be IPv4 or IPv6; enter a subnet mask (for IPv4) or prefix length (for IPv6).

`#(config service_name) view`
Views the specified proxy service.

For More Information

- *SGOS Administration Guide*

Example

```
SGOS#(config proxy-services) create https-reverse-proxy HTTPS_RP1
SGOS#(config proxy-services) edit HTTPS_RP1
SGOS#(config HTTPS_RP1) attribute use-adn enable
ok
```

#(config mms)

Synopsis

Allows you to manage services that are controlled by the MMS proxy.

Syntax

```
#(config proxy-services) create service_type service_name
#(config proxy-services) edit service_name
```

This changes the prompt to:

```
#(config service_name)
```

Subcommands

`#(config service_name) add {all | source_ip | source_ip/subnet-mask} {transparent | explicit | all | destination_ip | destination_ip/subnet-mask} {port | first_port-last_port} [intercept | bypass]`
Allows you to add a listener with the parameters you specify.

`#(config service_name) bypass {all | source_ip | source_ip/subnet-mask} {transparent | explicit | all | destination_ip | destination_ip/subnet-mask} {port | first_port-last_port}`
Change the behavior from intercept to bypass for the listener you specify.

`#(config service_name) exit`
Exits to the `#(config proxy-services)` prompt.

`#(config service_name) group service-group`
Allows you to move a particular service to another service group.

`#(config service_name) intercept {all | source_ip | source_ip/subnet-mask} {transparent | explicit | all | destination_ip | destination_ip/subnet-mask} {port | first_port-last_port}`
Change the behavior from bypass to intercept for the listener you specify.

`#(config service_name) proxy-type proxy-type`
Allows you to change the proxy type of a particular service.

`#(config service_name) remove {all | source_ip | source_ip/subnet-mask} {transparent | explicit | all | destination_ip | destination_ip/subnet-mask} {port | first_port-last_port}`
Allows you to remove a listener with the parameters you specify.

`#(config service_name) view`
Views the specified proxy service.

For More Information

- *SGOS Administration Guide*

Example

```
SGOS#(config proxy-services) create mms mms1
SGOS#(config proxy-services) edit mms1
```

#(config msn-im)

Synopsis

Allows you to manage services that are controlled by the MSN-IM proxy.

Syntax

```
#(config proxy-services) create service_type service_name
#(config proxy-services) edit service_name
```

This changes the prompt to:

```
#(config service_name)
```

Subcommands

`#(config service_name) add {all | source_ip | source_ip/subnet-mask} {destination_ip | destination_ip/subnet-mask} {port | first_port-last_port} [intercept | bypass]`
Allows you to add a listener with the parameters you specify.

`#(config service_name) bypass {all | source_ip | source_ip/subnet-mask} {destination_ip | destination_ip/subnet-mask} {port | first_port-last_port}`
Changes the behavior from intercept to bypass for the listener you specify.

`#(config service_name) exit`
Exits to the `#(config proxy-services)` prompt.

`#(config service_name) group service-group`
Allows you to move a particular service to another service group.

`#(config service_name) intercept {all | source_ip | source_ip/subnet-mask} {destination_ip | destination_ip/subnet-mask} {port | first_port-last_port}`
Changes the behavior from bypass to intercept for the listener you specify.

`#(config service_name) proxy-type proxy-type`
Allows you to change the proxy type of a particular service.

`#(config service_name) remove {all | source_ip | source_ip/subnet-mask} {destination_ip | destination_ip/subnet-mask} {port | first_port-last_port}`
Allows you to remove a listener with the parameters you specify.

`#(config service_name) view`
Views the specified proxy service.

For More Information

- *SGOS Administration Guide*

Example

```
SGOS#(config proxy-services) create msn-im msn1
SGOS#(config proxy-services) edit msn1
```

#(config restricted-intercept)

Synopsis

By default, all clients and servers evaluate the entries in Proxy Services (**Configuration > Services > Proxy Services**) where the decision is made to intercept or bypass a connection. To restrict or reduce the clients and servers that can be intercepted by proxy services, use the restricted intercept list. The restricted intercept list is useful in a rollout, prior to full production, where you only want to intercept a subset of the clients. After you are in full production mode, the restricted intercept list can be disabled.

Enabling restricted intercept only intercepts traffic specified in the client/server list. Disabling restricted intercept results in normal interception.

Syntax

```
#(config) proxy-services
#(config proxy-services) restricted-intercept
```

The prompt changes to:

```
#(config restricted-intercept)
```

Subcommands

`#(config restricted-intercept) {enable | disable}`
Enables or disabled the restricted-intercept list.

`#(config restricted-intercept) add {all | client_ip | client_ip/subnet-mask} | {all| server_ip | server_ip/subnet-mask}`
Adds an entry to the restricted list, either a client or a server. IP addresses can be IPv4 or IPv6; enter a subnet mask (for IPv4) or prefix length (for IPv6).

`#(config restricted-intercept) remove {all | client_ip | client_ip/subnet-mask} | all | server_ip | server_ip/subnet-mask}`
Clears the specified client or server from the restricted list. IP addresses can be IPv4 or IPv6; enter a subnet mask (for IPv4) or prefix length (for IPv6).

`#(config restricted-intercept) view {<Enter> | filter {all | client_ip | client_ip/subnet-mask} | {all | server_ip | server_ip/subnet-mask}`
Allows you view the entire list or to filter on specific clients or servers. IP addresses can be IPv4 or IPv6; enter a subnet mask (for IPv4) or prefix length (for IPv6).

For More Information

- *SGOS Administration Guide*

Example

```
#(config) proxy-services
#(config proxy-services) restricted-intercept
#(config restricted-intercept) add all 192.168.100.1
```

#(config rtmp)

Synopsis

Allows you to manage services that are controlled by the RTMP proxy.

Syntax

```
#(config proxy-services) create service_type service_name
#(config proxy-services) edit service_name
```

This changes the prompt to:

```
#(config service_name)
```

Subcommands

`#(config service_name) add {all | source_ip | source_ip/subnet-mask} {transparent | explicit | all | destination_ip | destination_ip/subnet-mask} {port | first_port-last_port} [intercept | bypass]`
Allows you to add a listener with the parameters you specify.

`#(config service_name) bypass {all | source_ip | source_ip/subnet-mask} {transparent | explicit | all | destination_ip | destination_ip/subnet-mask} {port | first_port-last_port}`
Change the behavior from intercept to bypass for the listener you specify.

`#(config service_name) exit`
Exits to the `#(config proxy-services)` prompt.

`#(config service_name) group service-group`
Allows you to move a particular service to another service group.

`#(config service_name) intercept {all | source_ip | source_ip/subnet-mask} {transparent | explicit | all | destination_ip | destination_ip/subnet-mask} {port | first_port-last_port}`
Change the behavior from bypass to intercept for the listener you specify.

`#(config service_name) proxy-type proxy-type`
Allows you to change the proxy type of a particular service.

`#(config service_name) remove {all | source_ip | source_ip/subnet-mask} {transparent | explicit | all | destination_ip | destination_ip/subnet-mask} {port | first_port-last_port}`
Allows you to remove a listener with the parameters you specify.

`#(config service_name) view`
Views the specified proxy service.

For More Information

- *SGOS Administration Guide*

Example

```
SGOS#(config proxy-services) create rtmp rtsm1
SGOS#(config proxy-services) edit rtmp1
SGOS#(config rtsp1) proxy-type http
ok
```

#(config rtsp)

Synopsis

Allows you to manage services that are controlled by the RTSP proxy.

Syntax

```
#(config proxy-services) create service_type service_name
#(config proxy-services) edit service_name
```

This changes the prompt to:

```
#(config service_name)
```

Subcommands

`#(config service_name) add {all | source_ip | source_ip/subnet-mask} {transparent | explicit | all | destination_ip | destination_ip/subnet-mask} {port | first_port-last_port} [intercept | bypass]`
Allows you to add a listener with the parameters you specify.

`#(config service_name) bypass {all | source_ip | source_ip/subnet-mask} {transparent | explicit | all | destination_ip | destination_ip/subnet-mask} {port | first_port-last_port}`
Change the behavior from intercept to bypass for the listener you specify.

`#(config service_name) exit`
Exits to the `#(config proxy-services)` prompt.

`#(config service_name) group service-group`
Allows you to move a particular service to another service group.

`#(config service_name) intercept {all | source_ip | source_ip/subnet-mask} {transparent | explicit | all | destination_ip | destination_ip/subnet-mask} {port | first_port-last_port}`
Change the behavior from bypass to intercept for the listener you specify.

`#(config service_name) proxy-type proxy-type`
Allows you to change the proxy type of a particular service.

`#(config service_name) remove {all | source_ip | source_ip/subnet-mask} {transparent | explicit | all | destination_ip | destination_ip/subnet-mask} {port | first_port-last_port}`
Allows you to remove a listener with the parameters you specify.

`#(config service_name) view`
Views the specified proxy service.

For More Information

- *SGOS Administration Guide*

Example

```
SGOS#(config proxy-services) create rtsp rtsp1
SGOS#(config proxy-services) edit rtsp1
SGOS#(config rtsp1) attribute use-adn enable
ok
```

#(config socks)

Synopsis

Allows you to manage services that are controlled by the SOCKS proxy.

Syntax

```
#(config proxy-services) create service_type service_name
#(config proxy-services) edit service_name
```

This changes the prompt to:

```
#(config service_name)
```

Subcommands

`#(config service_name) add {all | source_ip | source_ip/subnet-mask} {explicit | destination_ip | destination_ip/subnet-mask} {port | first_port-last_port} [intercept | bypass]`
Allows you to add a listener with the parameters you specify.

`#(config service_name) attribute detect-protocol {disable | enable}`
Detects the protocol being used. Protocols that can be detected include: HTTP, P2P (eDonkey, BitTorrent, FastTrack, Gnutella), SSL, and Endpoint Mapper.

`#(config service_name) bypass {all | source_ip | source_ip/subnet-mask} {explicit | destination_ip | destination_ip/subnet-mask} {port | first_port-last_port}`
Change the behavior from intercept to bypass for the listener you specify..

`#(config service_name) exit`
Exits to the `#(config proxy-services)` prompt.

`#(config service_name) group service-group`
Allows you to move a particular service to another service group.

`#(config service_name) intercept {all | source_ip | source_ip/subnet-mask} {explicit | destination_ip | destination_ip/subnet-mask} {port | first_port-last_port}`
Change the behavior from bypass to intercept for the listener you specify.

`#(config service_name) proxy-type proxy-type`
Allows you to change the proxy type of a particular service.

`#(config service_name) remove {all | source_ip | source_ip/subnet-mask} {explicit | destination_ip | destination_ip/subnet-mask} {port | first_port-last_port}`
Allows you to remove a listener with the parameters you specify.

`#(config service_name) view`
Views the specified proxy service.

For More Information

- *SGOS Administration Guide*

Example

```
SGOS#(config proxy-services) create socks socks1
SGOS#(config proxy-services) edit socks1
SGOS#(config socks1) attribute detect-protocol enable
ok
```

#(config ssl)

Synopsis

Allows you to manage services that are controlled by the SSL proxy.

Syntax

```
#(config proxy-services) create service_type service_name
#(config proxy-services) edit service_name
```

This changes the prompt to:

```
#(config service_name)
```

Subcommands

`#(config service_name) add {all | source_ip | source_ip/subnet-mask} {transparent | explicit | all | destination_ip | destination_ip/subnet-mask} {port | first_port-last_port} [intercept | bypass]`
Allows you to add a listener with the parameters you specify. IP addresses can be IPv4 or IPv6; enter a subnet mask (for IPv4) or prefix length (for IPv6). When the destination address on a listener is defined as **transparent**, **explicit**, or **all**, it applies to IPv4 and IPv6 addresses.

`#(config service_name) attribute use-adn {disable | enable}`
Controls whether ADN is enabled for a specific service. Enabling ADN does not guarantee the connections are accelerated by ADN. The actual decision is determined by ADN routing (for explicit deployment) and network setup (for transparent deployment).

`#(config service_name) attribute adn-byte-cache {disable | enable}`
Controls whether to optimize traffic using the byte caching optimization technique when connecting upstream in an ADN tunnel.

`#(config service_name) attribute adn-compress {disable | enable}`
Controls whether to optimize traffic using GZIP compression when connecting upstream in an ADN tunnel.

`#(config service_name) attribute adn-optimize {disable | enable}`
Starting in SGOS 6.2, the `adn-optimize` command was replaced by the `adn-byte-cache` and `adn-compress` commands (see above).

`#(config service_name) attribute byte-cache-priority {low | normal | high}`
Adjust retention priority of byte cache data. If you want to keep certain types of streams in the byte cache for as long as possible, set a high retention priority for the service. Or for streams that aren't likely to get much benefit from byte caching, you can set a low retention priority for the related service. Most services are set to normal priority by default. Note that unless the underlying service has `adn-byte-cache` enabled, the priority setting will have no effect; if you try to set a retention priority when byte caching is disabled, a warning message displays to inform you that the `byte-cache-priority` attribute has no effect when `adn-byte-cache` is disabled.

`#(config service_name) bypass {all | source_ip | source_ip/subnet-mask} {transparent | explicit | all | destination_ip | destination_ip/subnet-mask} {port | first_port-last_port}`
Change the behavior from intercept to bypass for the listener you specify. IP addresses can be IPv4 or IPv6; enter a subnet mask (for IPv4) or prefix length (for IPv6). When the destination address on a listener is defined as **transparent**, **explicit**, or **all**, it applies to IPv4 and IPv6 addresses.

`#(config service_name) attribute detect_protocol {disable | enable}`
Controls whether to detect HTTPS protocol after intercepting the SSL traffic, and hand it off to the HTTPS proxy. Non-HTTPS traffic will be tunneled using STunnel.

`#(config service_name)` **`exit`**
Exits to the `#(config proxy-services)` prompt.

`#(config service_name)` **`group`** *`service-group`*
Moves the service to a different service group.

`#(config service_name)` **`intercept`** {**`all`** | *`source_ip`* | *`source_ip/subnet-mask`*} {`transparent` | `explicit` | `all` | *`destination_ip`* | *`destination_ip/subnet-mask`*} {*`port`* | *`first_port-last_port`*}
Change the behavior from bypass to intercept for the listener you specify. IP addresses can be IPv4 or IPv6; enter a subnet mask (for IPv4) or prefix length (for IPv6). When the destination address on a listener is defined as **`transparent`**, **`explicit`**, or **`all`**, it applies to IPv4 and IPv6 addresses.

`#(config service_name)` **`proxy-type`** *`proxy-type`*
Changes the proxy type.

`#(config service_name)` **`remove`** {**`all`** | *`source_ip`* | *`source_ip/subnet-mask`*} {`transparent` | `explicit` | `all` | *`destination_ip`* | *`destination_ip/subnet-mask`*} {*`port`* | *`first_port-last_port`*}
Removes a listener. IP addresses can be IPv4 or IPv6; enter a subnet mask (for IPv4) or prefix length (for IPv6).

`#(config service_name)` **`view`**
Views the specified proxy service.

For More Information

- *SGOS Administration Guide*

Example

```
SGOS#(config proxy-services) create ssl ssl1
SGOS#(config proxy-services) edit ssl1
SGOS#(config ssl1) add transparent 443
```

#(config static-bypass)

Synopsis

Static bypass prevents the ProxySG from transparently accelerating requests to servers that perform IP authentication with clients. When a request matches an IP address and subnet mask specification, the request is sent to the designated gateway without going through the ProxySG.

Syntax

```
#(config) proxy-services
#(config proxy-services) static-bypass
#(config static-bypass)
```

Subcommands

`#(config static-bypass) add {all | client_ip_address | client_ip_address/subnet-mask} {all | server_ip_address | server_ip_address/subnet-mask} ["<comment>"]`

Allows you to add a listener with the parameters you specify. IP addresses can be in IPv4 or IPv6; enter a subnet mask (for IPv4) or prefix length (for IPv6). *All* includes IPv4 and IPv6 addresses. You can optionally enter a comment of up to 80 characters enclosed in quotation marks (" ") that specifies why you want the specific source/destination combination to be bypassed. Adding a comment is also useful if another administrator needs to make changes to the configuration later.

`#(config static-bypass) exit`

Exits from the `#(config static-bypass)` mode and returns to the `#(config proxy-services)` mode.

`#(config static-bypass) remove {all | client_ip_address | client_ip_address/subnet-mask} {all | server_ip_address | server_ip_address/subnet-mask}`

Allows you to remove a listener with the parameters you specify. IP addresses can be in IPv4 or IPv6; enter a subnet mask (for IPv4) or prefix length (for IPv6). includes IPv4 and IPv6 addresses.

`#(config static-bypass) view {filter {* | all | client_ip_address | client_ip_address/ subnet-mask} {* | all | server_ip_address | server_ip_address/ subnet-mask}} | <Enter>}`

Allows you to view static bypass entries based on the filters you specify. IP addresses can be in IPv4 or IPv6; enter a subnet mask (for IPv4) or prefix length (for IPv6).

For More Information

- *SGOS Administration Guide*

Example

```
SGOS#(config proxy-services) static-bypass
SGOS #(config static-bypass) add 10.9.17.135 all
ok
```

#(config tcp-tunnel)

Synopsis

Allows you to manage services that are controlled by the TCP Tunnel proxy.

Syntax

```
#(config proxy-services) create service_type service_name [service_group]
#(config proxy-services) edit service_name
```

This changes the prompt to:

```
#(config service_name)
```

Subcommands

`#(config service_name) add {all | source_ip | source_ip/subnet-mask} {transparent | explicit | all | destination_ip | destination_ip/subnet-mask} {port | first_port-last_port} [intercept | bypass]`
Allows you to add a listener with the parameters you specify. IP addresses can be IPv4 or IPv6; enter a subnet mask (for IPv4) or prefix length (for IPv6). When the destination address on a listener is defined as **transparent**, **explicit**, or **all**, it applies to IPv4 and IPv6 addresses.

`#(config service_name) attribute use-adn {disable | enable}`
Controls whether ADN is enabled for a specific service. Enabling ADN does not guarantee the connections are accelerated by ADN. The actual decision is determined by ADN routing (for explicit deployment) and network setup (for transparent deployment).

`#(config service_name) attribute adn-byte-cache {disable | enable}`
Controls whether to optimize traffic using the byte caching optimization technique when connecting upstream in an ADN tunnel.

`#(config service_name) attribute adn-compress {disable | enable}`
Controls whether to optimize traffic using GZIP compression when connecting upstream in an ADN tunnel.

`#(config service_name) attribute adn-optimize {disable | enable}`
Starting in SGOS 6.2, the `adn-optimize` command was replaced by the `adn-byte-cache` and `adn-compress` commands (see above).

`#(config service_name) attribute byte-cache-priority {low | normal | high}`
Adjust retention priority of byte cache data. If you want to keep certain types of streams in the byte cache for as long as possible, set a high retention priority for the service. Or for streams that aren't likely to get much benefit from byte caching, you can set a low retention priority for the related service. Most services are set to normal priority by default. Note that unless the underlying service has `adn-byte-cache` enabled, the priority setting will have no effect; if you try to set a retention priority when byte caching is disabled, a warning message displays to inform you that the `byte-cache-priority` attribute has no effect when `adn-byte-cache` is disabled.

`#(config service_name) attribute detect-protocol {disable | enable}`
Detects the protocol being used. Protocols that can be detected include: HTTP, P2P (eDonkey, BitTorrent, FastTrack, Gnutella), SSL, and Endpoint Mapper.

`#(config service_name) attribute early-intercept {disable | enable}`
Controls whether the proxy responds to client TCP connection requests before connecting to the upstream server. When early intercept is disabled, the proxy delays responding to the client until after it has attempted to contact the server.

`#(config service_name) attribute adn-thin-client {disable | enable}`
Applies special treatment to streams from thin client applications (such as RDP, VNC, and Citrix). This processing improves responsiveness of thin client actions. For example, end users will notice that the

desktop displays significantly faster. This option is available only for services using the TCP Tunnel proxy, and can be enabled only when ADN is enabled and byte caching and/or compression is enabled. The `byte-cache-priority` and `adn-thin-client` settings are mutually exclusive; you cannot enable both options for a service.

`#(config` *service_name*`)` **`bypass`** `{`**`all |`** *source_ip* `|` *source_ip/subnet-mask*`}` `{transparent | explicit | all |` *destination_ip* `|` *destination_ip/subnet-mask*`}` `{`*port* `|` *first_port-last_port*`}`
Change the behavior from intercept to bypass for the listener you specify. IP addresses can be IPv4 or IPv6; enter a subnet mask (for IPv4) or prefix length (for IPv6). When the destination address on a listener is defined as **`transparent`**, **`explicit`**, or **`all`**, it applies to IPv4 and IPv6 addresses.

`#(config` *service_name*`)` **`exit`**
Exits to the `#(config proxy-services)` prompt.

`#(config` *service_name*`)` **`group`** *service-group*
Moves the service to a different service group.

`#(config` *service_name*`)` **`intercept`** `{`**`all |`** *source_ip* `|` *source_ip/subnet-mask*`}` `{transparent | explicit | all |` *destination_ip* `|` *destination_ip/subnet-mask*`}` `{`*port* `|` *first_port-last_port*`}`
Change the behavior from bypass to intercept for the listener you specify. IP addresses can be IPv4 or IPv6; enter a subnet mask (for IPv4) or prefix length (for IPv6). When the destination address on a listener is defined as **`transparent`**, **`explicit`**, or **`all`**, it applies to IPv4 and IPv6 addresses.

`#(config` *service_name*`)` **`proxy-type`** *proxy-type*
Changes the proxy type.

`#(config` *service_name*`)` **`remove`** `{`**`all |`** *source_ip* `|` *source_ip/subnet-mask*`}` `{transparent | explicit | all |` *destination_ip* `|` *destination_ip/subnet-mask*`}` `{`*port* `|` *first_port-last_port*`}`

`#(config` *service_name*`)` **`view`**
Views the specified proxy service.

For More Information

- *SGOS Administration Guide*

Example

```
SGOS#(config proxy-services) create tcp-tunnel TCP1
SGOS#(config proxy-services) edit TCP1
SGOS#(config TCP1) attribute early-intercept enable
ok
```

#(config telnet)

Synopsis

Allows you to manage services that are controlled by the Telnet proxy.

Syntax

```
#(config proxy-services) create service_type service_name
#(config proxy-services) edit service_name
```

This changes the prompt to

```
#(config service_name)
```

Subcommands

`#(config service_name) add {all | source_ip | source_ip/subnet-mask} {transparent | explicit | all | destination_ip | destination_ip/subnet-mask} {port | first_port-last_port} [intercept | bypass]`

Allows you to add a listener with the parameters you specify. IP addresses can be IPv4 or IPv6; enter a subnet mask (for IPv4) or prefix length (for IPv6). When the destination address on a listener is defined as **`transparent`**, **`explicit`**, or **`all`**, it applies to IPv4 and IPv6 addresses.

`#(config service_name) attribute use-adn {disable | enable}`

Controls whether ADN is enabled for a specific service. Enabling ADN does not guarantee the connections are accelerated by ADN. The actual decision is determined by ADN routing (for explicit deployment) and network setup (for transparent deployment).

`#(config service_name) attribute adn-byte-cache {disable | enable}`

Controls whether to optimize traffic using the byte caching optimization technique when connecting upstream in an ADN tunnel.

`#(config service_name) attribute adn-compress {disable | enable}`

Controls whether to optimize traffic using GZIP compression when connecting upstream in an ADN tunnel.

`#(config service_name) attribute adn-optimize {disable | enable}`

Starting in SGOS 6.2, the `adn-optimize` command was replaced by the `adn-byte-cache` and `adn-compress` commands (see above).

`#(config service_name) attribute byte-cache-priority {low | normal | high}`

Adjust retention priority of byte cache data. If you want to keep certain types of streams in the byte cache for as long as possible, set a high retention priority for the service. Or for streams that aren't likely to get much benefit from byte caching, you can set a low retention priority for the related service. Most services are set to normal priority by default. Note that unless the underlying service has `adn-byte-cache` enabled, the priority setting will have no effect; if you try to set a retention priority when byte caching is disabled, a warning message displays to inform you that the `byte-cache-priority` attribute has no effect when `adn-byte-cache` is disabled.

`#(config service_name) attribute detect-protocol {disable | enable}`

Detects the protocol being used. Protocols that can be detected include: HTTP, P2P (eDonkey, BitTorrent, FastTrack, Gnutella), SSL, and Endpoint Mapper.

`#(config service_name) attribute early-intercept {disable | enable}`

Controls whether the proxy responds to client TCP connection requests before connecting to the upstream server. When early intercept is disabled, the proxy delays responding to the client until after it has attempted to contact the server.

`#(config service_name) bypass {all | source_ip | source_ip/subnet-mask} {transparent | explicit | all | destination_ip | destination_ip/subnet-mask} {port | first_port-last_port}`

Change the behavior from intercept to bypass for the listener you specify. IP addresses can be IPv4 or IPv6; enter a subnet mask (for IPv4) or prefix length (for IPv6). When the destination address on a listener is defined as **transparent**, **explicit**, or **all**, it applies to IPv4 and IPv6 addresses.

`#(config service_name)` **exit**
Exits to the `#(config proxy-services)` prompt.

`#(config service_name)` **group** *service-group*
Moves the service to a different service group.

`#(config service_name)` **intercept** {**all |** *source_ip* | *source_ip/subnet-mask*} {transparent | explicit | all | *destination_ip* | *destination_ip/subnet-mask*} {*port* | *first_port-last_port*}
Change the behavior from bypass to intercept for the listener you specify. IP addresses can be IPv4 or IPv6; enter a subnet mask (for IPv4) or prefix length (for IPv6). When the destination address on a listener is defined as **transparent**, **explicit**, or **all**, it applies to IPv4 and IPv6 addresses.

`#(config service_name)` **proxy-type** *proxy-type*
Changes the proxy type.

`#(config service_name)` **remove** {**all |** *source_ip* | *source_ip/subnet-mask*} {transparent | explicit | all | *destination_ip* | *destination_ip/subnet-mask*} {*port* | *first_port-last_port*}

`#(config service_name)` **view**
Views the specified proxy service.

For More Information

- *SGOS Administration Guide*

Example

```
SGOS#(config proxy-services) create telnet telnet1
SGOS#(config proxy-services) edit telnet1
SGOS #(config telnet1) view
Service Name:   telnet1
Proxy:          Telnet
Attributes:     early-intercept
Destination IP    Port Range        Action
```

#(config yahoo-im)

Synopsis

Allows you to manage services that are controlled by the Yahoo IM proxy.

Syntax

```
#(config proxy-services) create service_type service_name
#(config proxy-services) edit service_name
```

This changes the prompt to:

```
#(config service_name)
```

Subcommands

`#(config service_name) add {all | source_ip | source_ip/subnet-mask} {destination_ip | destination_ip/subnet-mask} {port | first_port-last_port} [intercept | bypass]`
Allows you to add a listener with the parameters you specify.

`#(config service_name) bypass {all | source_ip | source_ip/subnet-mask} {destination_ip | destination_ip/subnet-mask} {port | first_port-last_port}`
Changes the behavior from intercept to bypass for the listener you specify. IP addresses can be IPv4 or IPv6; enter a subnet mask (for IPv4) or prefix length (for IPv6).

`#(config service_name) exit`
Exits to the `#(config proxy-services)` prompt.

`#(config service_name) group service-group`
Moves the service to a different service group.

`#(config service_name) intercept {all | source_ip | source_ip/subnet-mask} {destination_ip | destination_ip/subnet-mask} {port | first_port-last_port}`
Changes the behavior from bypass to intercept for the listener you specify.

`#(config service_name) proxy-type proxy-type`
Changes the proxy type.

`#(config service_name) remove {all | source_ip | source_ip/subnet-mask} {destination_ip | destination_ip/subnet-mask} {port | first_port-last_port}`

`#(config service_name) view`
Views the specified proxy service.

For More Information

- *SGOS Administration Guide*

Example

```
SGOS#(config proxy-services) create yahoo-im yahoo1
SGOS#(config proxy-services) edit yahoo1
```

#(config) restart

Synopsis

Use this command to set restart options for the ProxySG.

Syntax

`#(config)` **`restart core-image {context | full | keep`** *`number`* **`| none}`**

context: Indicates only core image context should be written on restart.
full: Indicates full core image should be written on restart.
keep *`numbers`*: Specifies a number of core images to keep on restart.
none: Indicates no core image should be written on restart.

`#(config)` **`restart mode {hardware | software}`**

hardware: Specifies a hardware restart.
software: Specifies a software restart.

For More Information

- *SGOS Administration Guide*

Example

```
SGOS#(config) restart mode software
 ok
```

#(config) return-to-sender

Synopsis

The Return-to-Sender (RTS) option eliminates the need to create static routes by configuring the ProxySG to send response packets back to the same interface that received the request packet, entirely bypassing any routing lookup on the ProxySG. Essentially, the ProxySG stores the source Ethernet MAC address that the client's packet came from and sends all responses to that address.Under these conditions, if the return-to-sender feature is enabled, the ProxySG remembers the MAC address of the last hop for a packet from the client or server and sends any responses or requests to the MAC address instead of the default gateway.

Inbound RTS affects connections initiated to the ProxySG by clients and is enabled by default in SGOS 5.4 and later. Inbound RTS configures the ProxySG to send SYN-ACK packets to the same interface that the SYN packet arrived on. All subsequent TCP/IP response packets are also sent to the same interface that received the request packet.

RTS inbound applies only to clients who are on a different subnet than the ProxySG. If clients are on the same subnet, interface routes are used.

Outbound RTS affects connections initiated by the ProxySG to origin servers. Outbound RTS causes the ProxySG to send ACK and subsequent packets to the same interface that the SYN-ACK packet arrived on.

Note: Return-to-sender functionality should only be used if static routes cannot be defined for the clients and servers or if routing information for the clients and servers is not available through RIP packets.

Load balancing: You can use inbound RTS for load balancing. Normally, the ProxySG would not know which load balancer to return the packet to. When inbound RTS is enabled, the ProxySG simply returns packets to the load balancer the packets came from.

Syntax

`#(config) return-to-sender inbound {disable | enable}`
Enables or disables return-to-sender for inbound sessions.

`#(config) return-to-sender outbound {disable | enable}`
Enables or disables return-to-sender for outbound sessions.

`#(config) return-to-sender overwrite-static-route {disable | enable}`
When enabled, `return-to-sender` will overwrite any static route entries. The default is disabled.

Example

```
SGOS#(config) return-to-sender inbound enable
 ok
```

#(config) reveal-advanced

- `# reveal-advanced` on page 77.

#(config) rip

Synopsis

Use this command to set RIP (Routing Information Protocol) configuration options.

Using RIP, a host and router can send a routing table list of all other known hosts to its closest neighbor host every 30 seconds. The neighbor host passes this information on to its next closest neighbor and so on until all hosts have perfect knowledge of each other. (RIP uses the hop count measurement to derive network distance.) Each host in the network can then use the routing table information to determine the most efficient route for a packet.

The RIP configuration is defined in a configuration file. To configure RIP, first create a text file of RIP commands and then load the file by using the `load` command.

Syntax

`#(config)` **`rip disable`**
Disables the current RIP configuration.

`#(config)` **`rip enable`**
Enables the current RIP configuration.

`#(config)` **`rip default-route {enable | disable}`**
Accepts or denies the incoming default route advertisement.

`#(config)` **`rip default-route`** `{`**`group`** *`number`* `|` **`weight`** *`number`*`}`
Allows you to set the preference group and weight of the default routes.

`#(config)` **`rip no path`**
Clears the current RIP configuration path as determined using the `rip path` *`url`* command.

`#(config)` **`rip path`** *`url`*
Sets the path to the RIP configuration file to the URL indicated by *`url`*.

For More Information

- *SGOS Administration Guide*

Example

```
SGOS#(config) rip path 10.25.36.47/files/rip.txt
 ok
```

#(config) security

The `#(config) security` command is used for security, authentication, and authorization. The security command, by itself, cannot be used. You must use `security` commands with the options discussed in Subcommands below.

Synopsis

The ProxySG provides the ability to authenticate and authorize explicit and transparent proxy users using industry-standard authentication services.

Syntax

```
#(config) security [subcommands]
```

Subcommands

Modes in the security command are divided into three categories:

- Console Access and Authorization
- Realms
- Transparent Proxy

Note: While the commands are listed in functional order below, they are discussed in alphabetical order in the pages that follow. Each of the options in blue are hyperlinked so you can go directly to the command.

Console Access and Authorization

The options in this category do not enter a new submode. These options allow you to manage passwords and usernames for the ProxySG itself.

`#(config) security allowed-access` on page 313
Adds or removes the specified IP address to the access control list.

`#(config) security default-authenticate-mode` on page 321
Sets the default `authenticate.mode` to `auto` or to `sg2.`

`#(config) security destroy-old-passwords` on page 322
Destroys recoverable passwords in configuration used by previous versions.

`#(config) security enable-password and hashed-enable-password` on page 323
Sets the console enable password to the password specified.

`#(config) security encrypted-enable-password` on page 324
Specify an encrypted console enable password.

`#(config) security encrypted-password` on page 325
Specify an encrypted console account password.

`#(config) security enforce-acl` on page 326
Enables or disables the console access control list.

`#(config) security front-panel-pin and hashed-front-panel-pin` on page 327
Sets a four-digit PIN to restrict access to the front panel of the ProxySG.

`#(config) security legacy-relative-usernames` on page 328
Enables and disables the use of legacy relative usernames.

#(config) security management on page 343
Manages display settings.

#(config) security password and hashed_password on page 346
Specifies the console enable password in hashed format.

#(config) security password-display on page 347
Specifies format to display passwords in `show config` output.

#(config) security users on page 368
Manages user log ins, log outs and refresh data

#(config) security username on page 369
Specifies the console username.

Realms

Multiple authentication realms can be used on a single ProxySG. Multiple realms are essential if the enterprise is a managed provider or the company has merged with or acquired another company. Even for companies using only one protocol, multiple realms might be necessary, such as the case of a company using an LDAP server with multiple authentication boundaries. You can use realm sequencing to search the multiple realms all at one time.

Note: Up to 40 realms per type (such as certificate, authentication forms, and RADIUS) are allowed.

#(config) security authentication-forms on page 314
Creates forms for authentication and manage them.

#(config) security certificate on page 316
Creates and manages certificate realms.

#(config) security coreid on page 318
Creates and manages COREid realms.

#(config) security iwa-bcaaa on page 329
Creates and manages IWA realms that connect to Active Directory using BCAAA.

#(config) security iwa-direct on page 332
Creates and manages IWA realms that connect to Active Directory directly.

#(config) security ldap on page 335
Creates and manages LDAP realms.

#(config) security local on page 339
Creates and manages local realms.

#(config) security local-user-list on page 341
Creates and manages local user lists.

#(config) security novell-sso on page 344
Creates and manages Novell SSO realms.

#(config) security policy-substitution on page 348
Creates and manage policy-substitution realms.

#(config) security radius on page 351
Creates and manages RADIUS realms.

#(config) security request-storage on page 354
Creates and manages request-storage realms.

#(config security sequence) on page 360
Creates and manages sequence realms.

#(config) security siteminder on page 362
Creates and manages SiteMinder realms.

#(config security windows-domains) on page 370
Configures a Windows domain for the encrypted MAPI feature.

#(config) security windows-sso on page 371
Creates and manages Windows SSO realms.

#(config) security xml on page 373
Creates and manages XML realms.

Transparent Proxy

You can configure the authentication method for transparent proxies.

#(config) security transparent-proxy-auth on page 366
Specifies certain transparent proxy authentication settings.

For More Information

- *SGOS Administration Guide*

Example

```
#(config) show security
Account:
   Username:        “admin”
   Hashed Password: $1$a2zTlEE$1b88R3SXUTXS.zO7lh8db0
   Hashed Enable Password: $1$xQnqGerX$LU65b20trsIAF6yJox26L.
   Hashed Front Panel PIN: "$1$ThSEiB1v$seyBhSxtTXEtUGDZ5NOB1/"
   Management console display realm name: "Aurora"
   Management console auto-logout timeout: Never
Access control is disabled
Access control list (source, mask):
Flush credentials on policy update is enabled
Default authenticate.mode: auto
Transparent proxy authentication:
  Method: cookie
  Cookie type: session
  Cookie virtual-url: "www.cfauth.com/"
  IP time-to-live: 15
Local realm:
  No local realm is defined.
RADIUS realm:
  No RADIUS realm is defined.
LDAP realm(s):
  No LDAP realm is defined.
IWA realm(s):
  No IWA realm is defined.
Certificate realm(s):
  No certificate realms are defined.
SiteMinder realm(s):
  No realms defined.
COREid realm(s):
  No realms defined.
Policy-substitution realm(s):
  No realms defined.
Realm sequence(s):
  No realm sequences defined.
```

#(config) security allowed-access

Synopsis

Adds or removes IP addresses to the console access control list.

Syntax

```
#(config) security allowed-access [subcommands]
```

Subcommands

`#(config) security allowed-access add` *source_ip* [*ip_mask*]
Adds the specified IP address to the access control list.

`#(config) security allowed-access remove` *source_ip* [*ip_mask*]
Removes the specified IP from the access control list.

For More Information

- **#(config) security enforce-acl** on page 326
- *SGOS Administration Guide*

Example

```
#(config) security allowed-access add 10.25.36.47
```

#(config) security authentication-forms

You can use forms-based authentication exceptions to control what your users see during authentication. link.

To create and put into use forms-based authentication, you must complete the following steps:

- Create a new form or edit one of the existing authentication form exceptions
- Set storage options
- Set policies

Synopsis

Allows you to create and manage authentication forms.

Syntax

```
#(config) security authentication-forms
```

This changes the prompt to:

```
#(config authentication-forms)
```

Subcommands

`#(config authentication-forms)` **`copy`** [*source_form_name target_form_name*
Changes the name of a form. Note that you cannot change the form type.

`#(config authentication-forms)` **`create`** {**`authentication-form`** | **`new-pin-form`** | **`query-form`**} *form_name*
Creates a new authentication form using the form type you specify.

`#(config authentication-forms)` **`delete`** *form_name*
Deletes an authentication form

`#(config authentication-forms)` **`exit`**
Returns to the `#(config)` prompt.

`#(config authentication-forms)` **`inline`** *form_name eof_marker*
Installs an authentication form from console input.

`#(config authentication-forms)` **`load`** *form_name*
Downloads a new authentication form.

`#(config authentication-forms)` **`no path`** [*form_name*]
Negates authentication-form configuration.

`#(config authentication-forms)` **`path`** [*form_name*] *path*
Specifies the path (URL or IP address) from which to load an authentication form, or the entire set of authentication forms.

`#(config authentication-forms)` **`revert`** [*form_name*]
Reverts an authentication form to default.

`#(config authentication-forms)` **`view`**
Views the form specified or all forms.

For More Information

- **`#(config) security request-storage`** on page 354
- *SGOS Administration Guide*

Example

```
#(config) security authentication-forms
#(config authentication-forms) create form_type form_name
  ok
```

where *form_type* indicates the default *authentication-form, new-pin-form,* or *query-form* and *form_name* is the name you give the form.

#(config) security certificate

After an SSL session has been established, the user is asked to select the certificate to send to the ProxySG. If the certificate was signed by a Certificate Signing Authority that the ProxySG trusts, including itself, then the user is considered authenticated. The username for the user is the one extracted from the certificate during authentication.

You do not need to specify an authorization realm if:

- The policy does not make any decisions based on groups
- The policy works as desired when all certificate realm-authenticated users are not in any group

Synopsis

Allows you to create and manage certificate realms.

Syntax

```
#(config) security certificate [subcommands]
```

Subcommands

`#(config) security certificate create-realm` *realm_name*
Creates the specified certificate realm.

`#(config) security certificate delete-realm` *realm_name*
Deletes the specified certificate realm.

`#(config) security certificate edit-realm` *realm_name*
Changes the prompt. See Submodes for details.

`#(config) security certificate view` [*realm_name*]
Displays the configuration of all certificate realms or just the configuration for *realm_name* if specified.

Submodes

`#(config) security certificate edit-realm` *realm_name*

This changes the prompt to:

`#(config` *certificate_realm*`)`

Commands in this submode:

`#(config certificate` *certificate_realm*`) authorization ignore-user-list {add | clear | remove}`
Manages the ignore-user-list, which is the list of those to ignore if they are returned as search results.

`##(config certificate` *certificate_realm*`) authorization realm {none | realm-name` *realm_name*`}`
Specifies the authorization realm to use. Only LDAP, XML, and local realms are valid authorization realms.

`#(config certificate` *certificate_realm*`) authorization search-filter` *search_filter*
Specifies the search filter that should be used during a search of the LDAP server. The filter can contain policy substitutions including `$(cs-username)`.

`#(config certificate` *certificate_realm*`) authorization user-attribute {fqdn |` *LDAP_attribute_name*`}`
Specifies the user-attribute (fully qualified domain name or an LDAP attribute name) to be used during a search of the LDAP server.

`#(config certificate` *`certificate_realm`*`)` **`authorization username`** `{`**`determine-by-search`** `|` **`use-full-username`** `|` *`username_for_authorization`*`}`
Specifies the way a username should be determined. The default is the attribute `cn`, which specifies the user's relative name.

`#(config certificate` *`certificate_realm`*`)` **`cookie`** `{`**`persistent`** `{`**`enable`** `|` **`disable`**`}` `|` **`verify-ip`** `{`**`enable`**`|`**`disable`**`}`
Specifies whether to enable persistent or session cookies, and whether to verify the IP address of the cookie.

`#(config certificate` *`certificate_realm`*`)` **`display-name`** *`display_name`*
Specifies the display name for this realm.

`#(config certificate` *`certificate_realm`*`)` **`extended-key-usage`** `{`**`add`** `|` **`clear`** `|` **`remove`**`}`
Allows you to add and remove extended key usage OIDs and clear the OID list.

`#(config certificate` *`certificate_realm`*`)` **`exit`**
Exits #(`config` *`certificate_realm`*) mode and returns to (`config`) mode.

`#(config certificate` *`certificate_realm`*`)` **`identification full-username`** *`full username`*
Configures the syntax to extract the full username.

`#(config certificate` *`certificate_realm`*`)` **`identification username`** *`username`*
Configures the syntax to extract the username.

`#(config certificate` *`certificate_realm`*`)` **`inactivity-timeout`** *`seconds`*
Specifies the amount of time a session can be inactive before being logged out.

`#(config certificate` *`certificate_realm`*`)` **`refresh-time`** `{`**`authorization-refresh`** *`seconds`* `|` **`surrogate-refresh`** *`seconds`*`}`
Sets the refresh time for authorization and surrogates.

`#(config certificate` *`certificate_realm`*`)` **`rename`** *`new_realm_name`*
Renames this realm to *`new_realm_name`*.

`#(config certificate` *`certificate_realm`*`)` **`view`**
Displays this realm's configuration.

`#(config certificate` *`certificate_realm`*`)` **`virtual-url`** *`url`*
Specifies the virtual URL to use for this realm. If no URL is specified the global transparent proxy virtual URL is used.

For More Information

- **`#(config) security ldap`** on page 335
- **`#(config) security local`** on page 339
- **`#(config) security xml`** on page 373
- *SGOS Administration Guide*

Example

```
#(config) security certificate edit-realm testcert
#(config certificate testcert) refresh-time surrogate-refresh 800
 ok
#(config certificate testcert) exit
#(config)
```

#(config) security coreid

Within the COREid Access System, BCAAA acts as a custom AccessGate. It communicates with the COREid Access Servers to authenticate the user and to obtain a COREid session token, authorization actions, and group membership information.

Synopsis

Allows you to create and manage COREid realms.

Syntax

```
#(config) security coreid [subcommands]
```

Subcommands

`#(config)` **`security coreid create-realm`** *`realm_name`*
Creates the specified COREid realm

`#(config)` **`security coreid delete-realm`** *`realm_name`*
Deletes the specified COREid realm.

`#(config)` **`security coreid edit-realm`** *`realm_name`*
Changes the prompt. See Submodes for details.

`#(config)` **`security coreid view`** [*`realm_name`*]
Displays the configuration of all COREid realms or just the configuration for *`realm_name`* if specified.

Submodes

`#(config)` **`security coreid edit-realm`** *`realm_name`*

This changes the prompt to:

`#(config coreid` *`realm_name`*`)`

Commands in this submode:

`#(config coreid` *`realm_name`*`)` **`access-server-hostname`** *`hostname`*
The hostname of the primary Access Server.

`#(config coreid` *`realm_name`*`)` **`access-server-id`** ***`id`***
The ID of the primary Access Server.

`#(config coreid` *`realm_name`*`)` **`access-server-port`** *`port`*
The port of the primary Access Server

`#(config coreid` *`realm_name`*`)` **`add-header-responses disable | enable`**
When enabled, authorization actions from the policy domain obtained during authentication are added to each request forwarded by the ProxySG. Note that header responses replaces any existing header of the same name; if no such header exists, the header is added. Cookie responses replace a cookie header with the same cookie name; if no such cookie header exists, one is added.

`#(config coreid` *`realm_name`*`)` **`alternate-agent accessgate-id`** ***`name`***
The ID of the alternate AccessGate agent.

`#(config coreid` *`realm_name`*`)` **`alternate-agent encrypted-secret`**
`encrypted``_shared_secret`
The encrypted password associated with the alternate AccessGate. (Passwords can be up to 64 characters long and are always case sensitive.) The primary use of the encrypted-secret command is to allow the ProxySG to reload a password that it encrypted. If you choose to use a third-party encryption application, be sure it supports RSA encryption, OAEP padding, and is Base64 encoded with no newlines|

`#(config coreid` *realm_name)* **alternate-agent host** *hostname*
The hostname or the IP address of the alternate system that contains the agent.

`#(config coreid` *realm_name)* **alternate-agent port** *port*
The port where the alternate agent listens.

`#(config coreid` *realm_name)* **alternate-agent secret** *shared_secret*
The password associated with the alternate AccessGate. (Passwords can be up to 64 characters long and are always case sensitive.)

`#(config coreid` *realm_name)* **always-redirect-offbox** {**disable** | **enable**}
Forces authentication challenges to always be redirected to an off-box URL.

`#(config coreid` *realm_name)* **case-sensitive** {**disable** | **enable**}
Specifies whether the username and group comparisons on the ProxySG should be case-sensitive.

`#(config coreid` *realm_name)* **certificate-path** *certificate_path*
If Cert mode is used, the location on the BCAAA host machine where the key, server and CA chain certificates reside. The certificate files must be named aaa_key.pem, aaa_cert.pem and aaa_chain.pem respectively.

`#(config coreid` *realm_name*`)` **cookie** {**persistent** {**enable** | **disable**} | **verify-ip** {**enable** | **disable**}
Specifies whether to enable persistent or session cookies, and whether to verify the IP address of the cookie.

`#(config coreid` *realm_name)* **display-name** *display_name*
Equivalent to the display-name option in the CPL authenticate action. The default value for the display name is the realm name. The display name cannot be longer than 128 characters and it cannot be null.

`#(config coreid` *realm_name)* **encrypted-transport-pass-phrase** ***encrypted_pass_phrase***
If Simple or Cert mode is used, the Transport encrypted passphrase configured in the Access System.

`#(config coreid` *realm_name)* **exit**
Exits the `#(config coreid)` edit mode and returns to `#(config)` mode.

`#(config coreid` *realm_name*`)` **inactivity-timeout** *seconds*
Specifies the amount of time a session can be inactive before being logged out.

`#(config coreid` *realm_name*`)` **log-out** {**challenge** {**enable** | **disable**} | **display-time** *seconds*}
Allows you to challenge the user after log out and define the log out page display time.

`#(config coreid` *realm_name)* **no alternate-agent** | **certificate-path**
Removes the alternate agent configuration or the certificate path.

`#(config coreid` *realm_name)* **primary-agent accessgate-id** ***name***
The ID of the primary AccessGate agent.

`#(config coreid` *realm_name)* **primary-agent encrypted-secret** ***encrypted***_*shared_secret*
The encrypted password associated with the primary AccessGate. (Passwords can be up to 64 characters long and are always case sensitive.) The primary use of the encrypted-secret command is to allow the ProxySG to reload a password that it encrypted. If you choose to use a third-party encryption application, be sure it supports RSA encryption, OAEP padding, and is Base64 encoded with no new line.

`#(config coreid` *realm_name)* **primary-agent host** *hostname*
The hostname or the IP address of the primary system that contains the agent.

`#(config coreid` *realm_name)* **primary-agent port** *port*
The port where the primary agent listens.

`#(config coreid` *realm_name)* **primary-agent secret** *shared_secret*
The password associated with the primary AccessGate. (Passwords can be up to 64 characters long and are always case sensitive.)

`#(config coreid realm_name) protected-resource-name resource_name`
The resource name defined in the Access System policy domain

`#(config coreid realm_name) refresh-time {credential-refresh seconds | rejected-credentials-refresh seconds | surrogate-refresh seconds}`
Sets the refresh time for credential, rejected credentials cache, and surrogates.

`#(config coreid realm_name) rename new_realm_name`
Renames the realm to your request.

`#(config coreid realm_name) security-mode {cert | open | simple}`
The Security Transport Mode for the AccessGate to use when communicating with the Access System

`#(config coreid realm_name) ssl {disable | enable}`
Enable or disable SSL.

`#(config coreid realm_name) ssl-device-profile ssl_device_profile_name`
Specifies the device profile to use.

`#(config coreid realm_name) timeout seconds`
The length of time to elapse before timeout if a response from BCAAA is not received.

`#(config coreid realm_name) transport-pass-phrase pass_phrase`
If Simple or Cert mode is used, the Transport passphrase configured in the Access System.

`#(config coreid realm_name) validate-client-IP {disable | enable}`
Enables validation of the client IP address in SSO cookies. If the client IP address in the SSO cookie can be valid yet different from the current request client IP address due to downstream proxies or other devices, then disable client IP address validation. The WebGates participating in SSO with the ProxySG should also be modified. The WebGateStatic.lst file should be modified to either set the ipvalidation parameter to false or to add the downstream proxy/device to the IPValidationExceptions lists.

`#(config coreid realm_name) view`
Views the realm configuration.

`#(config coreid realm_name) virtual-url url`
The URL to redirect to when the user needs to be challenged for credentials. If the ProxySG is participating in SSO, the virtual hostname must be in the same cookie domain as the other servers participating in the SSO. It cannot be an IP address or the default.

For More Information

- **#(config) security siteminder** on page 362
- *SGOS Administration Guide*

Example

```
SGOS#(config) security coreid edit-realm coreid_1
SGOS#(config coreid coreid_1) access-server-hostname AccessServer_1
SGOS#(config coreid coreid_1) refresh-time surrogate-refresh 800
SGOS#(config coreid coreid_1) exit
```

#(config) security default-authenticate-mode

Synopsis

Sets the default `authenticate.mode` to `auto` or to `sg2`.

Syntax

```
#(config) security default-authenticate-mode [auto | sg2]
```

Subcommands

`#(config) security default-authenticate-mode auto`
Enables the access control list.

`#(config) security default-authenticate-mode sg2`
Disables the access control list.

For More Information

- *SGOS Administration Guide*

Example

```
SGOS#(config) security default-authenticate-mode auto
```

#(config) security destroy-old-passwords

Synopsis

Destroys recoverable passwords in configuration used by previous versions.

Syntax

```
#(config) security destroy-old-passwords [force]
```

Subcommands

`#(config) security destroy-old-passwords`
Destroys passwords after prompting.

`#(config) security destroy-old-passwords force`
Destroys passwords without prompting.

Note: Do not use this command if you intend to downgrade, as the old passwords are destroyed.

For More Information

- *SGOS Administration Guide*

Example

```
#(config) destroy-old-passwords force
```

#(config) security enable-password and hashed-enable-password

Synopsis

Sets the console enable password to the password specified.

Syntax

```
#(config) security enable-password password
#(config) security hashed-enable-password hashed_password
```

Subcommands

`#(config) security enable-password password | <enter>`
This is the password required to enter enable mode from the CLI when using console credentials, the serial console, or RSA SSH.

`#(config) security hashed-enable-password hashed_password`
The enable password in hashed format. You can either hash the password prior to entering it, or you can allow the ProxySG to hash the password.

For More Information

- *SGOS Administration Guide*

Example

```
#(config) security enable-password test
```

#(config) security encrypted-enable-password

Synopsis

Sets the console enable password to the encrypted password specified.

Syntax

```
#(config) security encrypted-enable-password [subcommand]
```

Subcommands

```
#(config) security encrypted-enable-password encrypted password
```
Sets the console enable password to the password specified.

For More Information

- *SGOS Administration Guide*

#(config) security encrypted-password

Synopsis

Sets the console account password to the encrypted password specified.

Syntax

```
#(config) security encrypted-password [subcommand]
```

Subcommands

```
#(config) security encrypted-password encrypted password
```
Sets the console account password to the password specified.

For More Information

- *SGOS Administration Guide*

#(config) security enforce-acl

Synopsis

Enables or disables the console access control list (ACL).

Syntax

```
#(config) security enforce-acl [enable | disable]
```

Subcommands

#(config) security enforce-acl enable
Enables the access control list.

#(config) security enforce-acl disable
Disables the access control list.

For More Information

- **#(config) alert** on page 116

Example

```
#(config) security enforce-acl disable
```

#(config) security front-panel-pin and hashed-front-panel-pin

Synopsis

Sets a four-digit PIN to restrict access to the front panel of the ProxySG.

Syntax

```
#(config) security front-panel-pin PIN
```

Subcommands

```
#(config) security front-panel-pin PIN
```
Use of this command is recommended for security reasons.

Note: To clear the PIN, specify 0000.

For More Information

- *SGOS Administration Guide*

Example

```
#(config) security front-panel-pin 1234
```

#(config) security legacy-relative-usernames

Synopsis

Enables and disables the use of legacy relative usernames.

Syntax

```
#(config) security legacy-relative-usernames [subcommands]
```

Subcommands

```
#(config) security legacy-relative-usernames {disable | enable}
```

Enables and disables use of legacy relative usernames.

For More Information

- *SGOS Administration Guide*

Example

```
#(config) security legacy-relative-usernames disable
 ok
```

#(config) security iwa-bcaaa

Integrated Windows Authentication (IWA) is an authentication mechanism available on Windows networks.

IWA is a Microsoft-proprietary authentication suite that allows Windows clients (running on Windows 2000 and higher) to automatically choose between using Kerberos and NTLM authentication challenge/response, as appropriate. When an IWA realm is used and a resource is requested by the client from the ProxySG appliance, the appliance contacts the client's domain account to verify the client's identity and request an access token. The access token is generated by the domain controller (in case of NTLM authentication) or a Kerberos server (in the case of Kerberos authentication) and passed to (and if valid, accepted by) the ProxySG appliance.

Refer to the Microsoft Web site for detailed information about the IWA protocol.

Synopsis

Allows you to create and manage IWA realms that connect to Active Directory using BCAAA.

Syntax

```
#(config) security iwa-bcaaa [subcommands]
```

Subcommands

`#(config) security iwa-bcaaa create-realm realm_name`
Creates the specified IWA realm.

`#(config) security iwa-bcaaa delete-realm realm_name`
Deletes the specified IWA realm.

`#(config) security iwa-bcaaa edit-realm realm_name`
Changes the prompt. See Submodes for details.

`#(config) security iwa-bcaaa view [realm_name]`
Displays the configuration of all IWA realms or just the configuration for *realm_name* if specified.

Submodes

```
#(config) security iwa-bcaaa edit-realm realm_name
```

This changes the prompt to:

```
#(config iwa-bcaaa realm_name)
```

Commands in this submode:

`#(config iwa-bcaaa realm_name) alternate-server host [port]`
Specifies the alternate server host and port.

`#(config iwa-bcaaa realm_name) cookie {persistent {enable | disable} | verify-ip {enable | disable}}`
Specifies whether to enable persistent or session cookies, and whether to verify the IP address of the cookie.

`#(config iwa-bcaaa realm_name) credentials-basic {disable | enable}`
Disables/enables support for Basic credentials in this realm. At least one of Basic or NTLM/Kerberos credentials must be supported.

`#(config iwa-bcaaa realm_name) credentials-kerberos {disable | enable}`
Disables/enables support for Kerberos credentials in this realm. If Kerberos is enabled, NTLM must also be enabled. At least one of Basic or NTLM/Kerberos credentials must be supported.

`#(config iwa-bcaaa realm_name) credentials-ntlm {disable | enable}`
Disables/enables support for NTLM credentials in this realm. If NTLM is enabled, Kerberos must also be enabled. At least one of Basic or NTLM/Kerberos credentials must be enabled.

`#(config iwa-bcaaa realm_name) display-name display_name`
Specifies the display name for this realm.

`#(config iwa-bcaaa realm_name) exit`
Exits the `iwa edit` mode and returns to (`config`) mode.

`#(config iwa-bcaaa realm_name) inactivity-timeout seconds`
Specifies the amount of time a session can be inactive before being logged out.

`#(config iwa-bcaaa realm_name) log-out {challenge {enable | disable} | display-time seconds}`
Allows you to challenge the user after log out and define the log out page display time.

`#(config iwa-bcaaa realm_name) no alternate-server`
Clears the alternate-server.

`#(config iwa-bcaaa realm_name) primary-server host [port]`
Specifies the primary server host and port.

`#(config iwa-bcaaa realm_name) refresh-time {credential-refresh seconds | rejected-credentials-refresh seconds | surrogate-refresh seconds}`
Sets the refresh time for credential, rejected credentials cache time, and surrogates.

`#(config iwa-bcaaa realm_name) rename new_realm_name`
Renames this realm to *new_realm_name*.

`#(config iwa-bcaaa realm_name) server-authentication {none | origin | proxy}`
Enables/disables the forwarding of BASIC credentials of the authenticated user to the origin content server or for proxy authentication. Flush the entries for a realm if the server-authentication value is changed to ensure that the server-authentication value is immediately applied.

You can only choose one server-authentication method:

- If set to **origin**, BASIC credentials are forwarded to an upstream server.
- If set to **proxy**, BASIC credentials are forwarded to an upstream proxy.
- If set to **none**, forwarding BASIC credentials is disabled.

`#(config iwa-bcaaa realm_name) ssl {disable | enable}`
Disables/enables SSL communication between the ProxySG and BCAAA.

`#(config iwa-bcaaa realm_name)ssl-device-profile ssl_device_profile_name`
Specifies the device profile to use.

`#(config iwa-bcaaa realm_name) test-authentication windows_domain_name\\username password`
Tests the IWA configuration to ensure that you can successfully authenticate a user in your Active Directory.

`#(config iwa-bcaaa realm_name) timeout seconds`
Specifies the IWA request timeout.

`#(config iwa-bcaaa realm_name) view`
Displays this realm's configuration.

`#(config iwa-bcaaa realm_name) virtual-url url`
Specifies the virtual URL to use for this realm. If no URL is specified the global transparent proxy virtual URL is used.

For More Information

- *SGOS Administration Guide*

Example

```
#(config) security iwa-bcaaa edit-realm testIWA
#(config iwa-bcaaa testIWA) no alternate server
 ok
#(config iwa-bcaaa testIWA) exit
#(config)
```

#(config) security iwa-direct

Integrated Windows Authentication (IWA) is an authentication mechanism available on Windows networks.

IWA is a Microsoft-proprietary authentication suite that allows Windows clients (running on Windows 2000 and higher) to automatically choose between using Kerberos and NTLM authentication challenge/response, as appropriate. When an IWA realm is used and a resource is requested by the client from the ProxySG appliance, the appliance contacts the client's domain account to verify the client's identity and request an access token. The access token is generated by the domain controller (in case of NTLM authentication) or a Kerberos server (in the case of Kerberos authentication) and passed to (and if valid, accepted by) the ProxySG appliance.

Refer to the Microsoft Web site for detailed information about the IWA protocol.

Synopsis

Allows you to create and manage IWA realms that allow the ProxySG appliance to connect directly to Active Directory.

Syntax

```
#(config) security iwa-direct [subcommands]
```

Subcommands

`#(config) security iwa-direct create-realm realm_name windows_domain_name`
Creates the specified IWA realm.

`#(config) security iwa-direct delete-realm realm_name`
Deletes the specified IWA realm.

`#(config) security iwa-direct edit-realm realm_name`
Changes the prompt. See Submodes for details.

`#(config) security iwa-direct view [realm_name]`
Displays the configuration of all IWA realms or just the configuration for *realm_name* if specified.

Submodes

```
#(config) security iwa-direct edit-realm realm_name
```

This changes the prompt to:

```
#(config iwa-direct realm_name)
```

Commands in this submode:

`#(config iwa-direct realm_name) cookie {persistent {enable | disable} | verify-ip {enable | disable}}`
Specifies whether to enable persistent or session cookies, and whether to verify the IP address of the cookie.

`#(config iwa-direct realm_name) credentials-basic {disable | enable}`
Disables/enables support for Basic credentials in this realm. At least one of Basic or NTLM/Kerberos credentials must be supported.

`#(config iwa-direct realm_name) credentials-kerberos {disable | enable}`
Disables/enables support for Kerberos credentials in this realm. If Kerberos is enabled, NTLM must also be enabled. At least one of Basic or NTLM/Kerberos credentials must be supported.

`#(config iwa-direct` *realm_name*`)` **`credentials-ntlm {disable | enable}`**
Disables/enables support for NTLM credentials in this realm. If NTLM is enabled, Kerberos must also be enabled. At least one of Basic or NTLM/Kerberos credentials must be enabled.

`#(config iwa-direct` *realm_name*`)` **`display-name`** *display_name*
Specifies the display name for this realm.

`#(config iwa-direct` *realm_name*`)` **`exit`**
Exits the `iwa edit` mode and returns to (`config`) mode.

`#(config iwa-direct` *realm_name*`)` **`inactivity-timeout`** *seconds*
Specifies the amount of time a session can be inactive before being logged out.

`#(config iwa-direct realm_name)` **`kerberos-user`** *username password*
Set the Kerberos User information needed for a load-balancing configuration

`#(config iwa-direct` *realm_name*`)` **`log-out {challenge {enable | disable} | display-time`** *seconds*`}`
Allows you to challenge the user after log out and define the log out page display time.

`#(config iwa-direct` *realm_name*`)` **`no kerberos-user`**
Clears the kerberos-user configuration.

`#(config iwa-direct` *realm_name*`)` **`refresh-time {credential-refresh`** *seconds* `|` **`rejected-credentials-refresh`** *seconds* `|` **`surrogate-refresh`** *seconds}*
Sets the refresh time for credential, rejected credentials cache time, and surrogates.

`#(config iwa-direct` *realm_name*`)` **`rename`** *new_realm_name*
Renames this realm to *new_realm_name*.

`#(config iwa-direct` *realm_name*`)` **`server-authentication {none | origin | proxy}`**
Enables/disables the forwarding of BASIC credentials of the authenticated user to the origin content server or for proxy authentication. Flush the entries for a realm if the server-authentication value is changed to ensure that the server-authentication value is immediately applied.

You can only choose one server-authentication method:

- If set to **`origin`**, BASIC credentials are forwarded to an upstream server.
- If set to **`proxy`**, BASIC credentials are forwarded to an upstream proxy.
- If set to **`none`**, forwarding BASIC credentials is disabled.

`#(config iwa-direct` *realm_name*`)` **`test-authentication`**
windows_domain_name`\\`*username password*
Tests the IWA configuration to ensure that you can successfully authenticate a user in your Active Directory.

`#(config iwa-direct` *realm_name*`)` **`timeout`** *seconds*
Specifies the IWA request timeout.

`#(config iwa-direct` *realm_name*`)` **`view`**
Displays this realm's configuration.

`#(config iwa-direct` *realm_name*`)` **`virtual-url`** *url*
Specifies the virtual URL to use for this realm. If no URL is specified the global transparent proxy virtual URL is used.

For More Information

- *SGOS Administration Guide*

Example

```
#(config) security iwa-direct edit-realm MyRealm
#(config iwa-direct MyRealm) virtual-url http://myproxy
 ok
#(config iwa-direct MyRealm) exit
#(config)
```

#(config) security ldap

Blue Coat supports both LDAP v2 and LDAP v3, but recommends LDAP v3 because it uses Transport Layer Security (TLS) and SSL to provide a secure connection between the ProxySG and the LDAP server.

An LDAP directory, either version 2 or version 3, consists of a simple tree hierarchy. An LDAP directory might span multiple LDAP servers. In LDAP v3, servers can return referrals to others servers back to the client, allowing the client to follow those referrals if desired.

Directory services simplify administration; any additions or changes made once to the information in the directory are immediately available to all users and directory-enabled applications, devices, and ProxySGs.

The ProxySG supports the use of external LDAP database servers to authenticate and authorize users on a per-group or per-attribute basis.

LDAP group-based authentication for the ProxySG can be configured to support any LDAP-compliant directory including:

- Microsoft Active Directory Server
- Novell NDS/eDirectory Server
- Netscape/Sun iPlanet Directory Server
- Other

Synopsis

Allows you to configure and manage LDAP realms.

Syntax

```
#(config) security ldap [subcommands]
```

Subcommands

```
#(config) security ldap create-realm realm_name
```
Creates the specified LDAP realm

```
#(config) security ldap delete-realm realm_name
```
Deletes the specified LDAP realm.

```
#(config) security ldap edit-realm realm_name
```
Changes the prompt. See Submodes for details.

```
#(config) security ldap view [realm_name]
```
Displays the configuration of all LDAP realms or just the configuration for *realm_name* if specified.

Submodes

```
#(config) security ldap edit-realm realm_name
```

This changes the prompt to:

```
#(config ldap realm_name)
```

Commands in the `ldap realm_name` mode:

```
#(config ldap realm_name) alternate-server host [port]
```
Specifies the alternate server host and port.

```
#(config ldap realm_name) case-sensitive {disable | enable}
```
Specifies whether or not the LDAP server is case-sensitive.

`#(config ldap` *realm_name*`)` **`cookie`** {**`persistent`** {**`enable`** | **`disable`**} | **`verify-ip`** {**`enable`** | **`disable`**}

Specifies whether to enable persistent or session cookies, and whether to verify the IP address of the cookie.

`#(config ldap` *realm_name*`)` **`default-group-name`** *default_group_name*

If the `validate-authorized-user` command is disabled and a default-group-name is configured, the default-group-name is used as the group name for non-existent users.

`#(config ldap` *realm_name*`)` **`display-name`** *display_name*

Specifies the display name for this realm.

`#(config ldap` *realm_name*`)` **`distinguished-name user-attribute-type`** *user_attribute_type*

Specifies the attribute type that defines the relative user name.

`#(config ldap` *realm_name*`)` **`distinguished-name base-dn`** {**`add`** | **`demote`** | **`promote`** | **`remove`**} {*base_dn* | **`clear`**}

Adds/demotes/promotes/removes a base DN from the base DN list, or clears the base DN list.

`#(config ldap` *realm_name*`)` **`exit`**

Exits the `ldap edit` mode and returns to `#(config)` mode.

`#(config ldap` *realm_name*`)` **`group-compare`** {**`local`** | **`server`**}

Specifies the method of LDAP group comparisons.

`#(config ldap` *realm_name*`)` **`group-search-constraint`** *ldap filter expression*

Adds an LDAP filter constraint to group searches.

`#(config ldap` *realm_name*`)` **`inactivity-timeout`** *seconds*

Specifies the amount of time a session can be inactive before being logged out.

`#(config ldap` *realm_name*`)` **`log-out`** {**`challenge`** {**`enable`** | **`disable`**} | **`display-time`** *seconds*}

Allows you to challenge the user after log out and define the log out page display time.

`#(config ldap` *realm_name*`)` **`lookup-group`** *simple_group_name*

Allows you to look up the common name of a group in your LDAP tree. For example, looking up the group `finance` might return a value such as `cn=finance,ou=headquarters,o=acme`.

`#(config ldap` *realm_name*`)` **`lookup-user`** *simple_user_name*

Allows you to look up the common name of a user in your LDAP tree. Note that this command will return all matching entries. For example, looking up the user `jdoe` might return a values such as `cn=jdoe,ou=headquarters,o=acme` and `CN=jdoe,o=acme`.

`#(config ldap` *realm_name*`)` **`membership-attribute`** *attribute_name*

Specifies the attribute that defines group membership.

`#(config ldap` *realm_name*`)` **`membership-type`** {**`group`** | **`user`**}

Specifies the membership type. Specify group if user memberships are specified in groups. Specify user if memberships are specified in users.

`#(config ldap` *realm_name*`)` **`membership-username (full`** | **`relative)`**

Specifies the username type to use during membership lookups. The `full` option specifies that the user's FQDN is used during membership lookups, and `relative` option specifies that the user's relative username is used during membership lookups. Only one can be selected at a time.

`#(config ldap` *realm_name*`)` **`nested-group-attribute`** *attribute_name*

Specifies the attribute that defines nested group membership. For **other**, **ad,** and **nds**, the default attribute name is **`member`**. For **iPlanet**, the default attribute name is **`uniqueMember.`**

`#(config ldap` *realm_name*`)` **`no {alternate-server`** | **`default-group-name`** | **`no membership-attribute`** | **`no nested-group-attribute`** | **`group-search-constraint}`**

Clears the attribute values.

`#(config ldap realm_name) objectclass container {add | remove} {container_objectclass | clear}`
Adds/removes container objectclass values from the list (these values are used during VPM searches of the LDAP realm), or clears all values from the container objectclass list.

`#(config ldap realm_name) objectclass group {add | remove} {group_objectclass | clear}`
Adds/removes group objectclass values from the list (these values are used during VPM searches of the LDAP realm), or clears all values from the group objectclass list.

`#(config ldap realm_name) objectclass user {add | remove} {user_objectclass | clear}`
Adds/removes user objectclass values from the list (these values are used during VPM searches of the LDAP realm), or clears all values from the user objectclass list.

`#(config ldap realm_name) primary-server host [port]`
Specifies the primary server host and port.

`#(config ldap realm_name) protocol-version {2 | 3}`
Specifies the LDAP version to use. SSL and referral processing are not available in LDAP v2.

`#(config ldap realm_name) referrals-follow {disable | enable}`
Disables/enables referral processing. This is available in LDAP v3 only.

`#(config ldap realm_name) refresh-time {authorization-refresh seconds | credential-refresh seconds | rejected-credentials-refresh seconds | surrogate-refresh seconds}`
Sets the refresh time for authorization, credential, rejected credentials cache, and surrogates.

`#(config ldap realm_name) rename new_realm_name`
Renames this realm to *new_realm_name*.

`#(config ldap realm_name) search anonymous {disable | enable}`
Disables/enables anonymous searches.

`#(config ldap realm_name) search dereference {always | finding | never | searching}`
Specifies the dereference level. Specify always to always dereference aliases. Specify finding to dereference aliases only while locating the base of the search. Specify searching to dereference aliases only after locating the base of the search. Specify never to never dereference aliases.

`#(config ldap realm_name) search encrypted-password encrypted_password`
Specifies the password to bind with during searches in encrypted format.

`#(config ldap realm_name) search password password`
Specifies the password to bind with during searches.

`#(config ldap realm_name) search user-dn user_dn`
Specifies the user DN to bind with during searches.

`#(config ldap realm_name) server-authentication {none | origin | proxy}`
Enables/disables the forwarding of BASIC credentials of the authenticated user to the origin content server or for proxy authentication. Flush the entries for a realm if the server-authentication value is changed to ensure that the server-authentication value is immediately applied.

You can only choose one server-authentication method:

- If set to **origin**, BASIC credentials are forwarded to an upstream server.
- If set to **proxy**, BASIC credentials are forwarded to an upstream proxy.
- If set to **none**, forwarding BASIC credentials is disabled.

`#(config ldap realm_name) server-type {ad | iplanet | nds | other}`
Specifies the LDAP server type for this realm.

`#(config ldap` *realm_name*`)` **`ssl {disable | enable}`**
Disables/enables SSL communication between the ProxySG and the LDAP server. This is only available in LDAP v3.

`#(config ldap` *realm_name*`)` **`ssl-device-profile`** *ssl_device_profile_name*
Specifies the device profile to use.

`#(config ldap` *realm_name*`)` **`support-nested-groups {disable | enable}`**
Enables or disables the nested group feature.

`#(config ldap` *realm_name*`)` **`test-authentication`** *username password*
Tests the LDAP configuration to ensure that the ProxySG can successfully authenticate a user in your LDAP realm using the username and password you provide.

`#(config ldap` *realm_name*`)` **`timeout`** *seconds*
Specifies the LDAP server's timeout.

`#(config ldap` *realm_name*`)` **`validate-authorized-user {enable | disable}`**
When `validate-authorized-user` is enabled, an **`authorization`** (not authentication) request verifies that the user exists in the LDAP server. If the user does not exist, the authorization request fails (authentication requests always require the user to exist).

When `validate-authorized-user` is disabled, no user existence check is made for an authorization request. If the user does not exist, the authorization request succeeds

`#(config ldap` *realm_name*`)` **`view`**
Displays this realm's configuration.

`#(config ldap` *realm_name*`)` **`virtual-url`** *url*
Specifies the virtual URL to use for this realm. If no URL is specified the global transparent proxy virtual URL is used.

For More Information

- *SGOS Administration Guide*

Example

```
#(config) security ldap edit-realm testldap
#(config ldap testldap) server-type iplanet
 ok
#(config ldap testldap) server-authentication origin
 ok
#(config ldap testldap) exit
```

#(config) security local

Using a Local realm is appropriate when the network topography does not include external authentication or when you want to add users and administrators to be used by the ProxySG only.

The Local realm (you can create up to 40) uses a *Local User List*, a collection of users and groups stored locally on the ProxySG. You can create up to 50 different Local User Lists. Multiple Local realms can reference the same list at the same time, although each realm can only reference one list at a time. The default list used by the realm can be changed at any time.

Synopsis

Allows you to configure and manage local realms.

Syntax

```
#(config) security local [subcommands]
```

Subcommands

`#(config) security local create-realm` *realm_name*
Creates the specified local realm.

`#(config) security local delete-realm` *realm_name*
Deletes the specified local realm.

`#(config) security local edit-realm` *realm_name*
Changes the prompt. See Submodes for details.

`#(config) security local view [`*realm_name*`]`
Displays the configuration of all local realms or just the configuration for *realm_name* if specified.

Submodes

```
#(config) security local edit-realm realm_name
```

This changes the prompt to:

```
#(config local realm_name)
```

Commands found in this submode include:

`#(config local` *realm_name*`) cookie {persistent {enable | disable} | verify-ip {enable | disable}`
Specifies whether to enable persistent or session cookies, and whether to verify the IP address of the cookie.

`#(config local` *realm_name*`) default-group-name` *default_group_name*
If the `validate-authorized-user` command is disabled and a default-group-name is configured, the default-group-name is used as the group name for non-existent users.

`#(config local` *realm_name*`) display-name` *display_name*
Specifies the display name for this realm.

`#(config local` *realm_name*`) exit`
Exits configure security local mode and returns to `#(config)` mode.

`#(config local` *realm_name*`) refresh-time {authorization-refresh` *seconds* `| surrogate-refresh` *seconds*`}`
Sets the refresh time for authorization and surrogates.

`#(config local` *realm_name*`) inactivity-timeout` *seconds*
Specifies the amount of time a session can be inactive before being logged out.

`#(config local realm_name) log-out {challenge {disable | enable} | display-time seconds}`
Configures the log-out behavior.

`#(config local realm_name) local-user-list local_user_list_name`
Specifies the local user list to for this realm.

`#(config local realm_name) no default-group-name`
Clears the default group name.

`#(config local realm_name) rename new_realm_name`
Renames this realm to *new_realm_name*

`#(config local realm_name) server-authentication {none | origin | proxy}`
Enables/disables the forwarding of BASIC credentials of the authenticated user to the origin content server or for proxy authentication. Flush the entries for a realm if the server-authentication value is changed to ensure that the server-authentication value is immediately applied.

You can only choose one server-authentication method:

- If set to **origin**, BASIC credentials are forwarded to an upstream server.
- If set to **proxy**, BASIC credentials are forwarded to an upstream proxy.
- If set to **none**, forwarding BASIC credentials is disabled.

`#(config local realm_name) validate-authorized-user {disable | enable}`
When `validate-authorized-user` is enabled, an **authorization** (not authentication) request verifies that the user exists in the local user list. If the user does not exist in the list, the authorization request fails (authentication requests always require the user to exist).

When `validate-authorized-user` is disabled, no user existence check is made for an authorization request. If the user does not exist, the authorization request succeeds.

`#(config local realm_name) view`
Displays this realm's configuration

`#(config local realm_name) virtual-url url`
Specifies the virtual URL to use for this realm. If no URL is specified the global transparent proxy virtual URL is used.

For More Information

- **#(config) security local-user-list** on page 341
- *SGOS Administration Guide*

Example

```
#(config) security local edit-realm testlocal
#(config local testlocal) server-authentication proxy
 ok
#(config local testlocal) exit
#(config)
```

#(config) security local-user-list

The local-user-list is only used in conjunction with local realms.

Synopsis

Manages the local-user-list used in local realms.

Syntax

`#(config) security local-user-list [subcommands]`

Subcommands

`#(config) security local-user-list clear [force]`
Clears all local user lists. Lists referenced by local realms and the default local user list are recreated but empty. Specify force to clear realms without a prompt for confirmation.

`#(config) security local-user-list create local-user-list`
Creates the local user list with the name specified

`#(config) security local-user-list default append-to-default {disable | enable}`
Disables/enables appending uploaded users to the default local user list.

`#(config) security local-user-list default list local_user_list`
Specifies the default local user list. The default list is populated during password file uploads. The default list is also the default list used by local realms when they are created

`#(config) security local-user-list delete local-user-list [force]`
Deletes the specified local user list. The default list and any lists used by local realms cannot be deleted. Specify `force` to delete the list without a prompt for confirmation.

`#(config) security local-user-list edit local-user-list`
Changes the prompt. See Submodes.

Submodes

`#(config) security local-user-list edit local_user_list`

This changes the prompt to:

`#(config local-user-list local_user_list)`

Commands found in this submode include:

`#(config local-user-list local_user_list) disable-all`
Disables all user accounts in the specified list.

`#(config local-user-list local_user_list) enable-all`
Enables all user accounts in the specified list.

`#(config local-user-list local_user_list) exit`
Exits configure local-user-list mode and returns to configure mode.

`#(config local-user-list local_user_list) group clear`
Clears all groups from the list. The users remain but do not belong to any groups.

`#(config local-user-list local_user_list) group create group_name`
Creates the specified group in the local user list.

`#(config local-user-list local_user_list) group delete group_name [force]`
Deletes the specified group in the local user list.

`#(config local-user-list local_user_list) lockout-duration seconds`
The length of time a user account is locked out after too many failed password attempts. The default is 3600

`#(config local-user-list` *`local_user_list`*`)` **`max-failed-attempts`** ***`attempts`***
The number of failed attempts to login to an ProxySG before the user account is locked. The default is 60 attempts.

`#(config local-user-list` *`local_user_list`*`)` **`no`** [**`lockout-duration`** | **`max-failed-attempts`** | **`reset-interval`**]
Disables the settings for this user list.

`#(config local-user-list` *`local_user_list`*`)` **`reset-interval`** *`seconds`*
The length of seconds to wait after the last failed attempt before resetting the failed counter to zero.

`#(config local-user-list` *`local_user_list`*`)` **`user clear`**
Clears all users from the list. The groups remain but do not have any users.

`#(config local-user-list` *`local_user_list`*`)` **`user create`** ***`user_name`***
Creates the specified user in the local user list.

`#(config local-user-list` *`local_user_list`*`)` **`user delete`** ***`user_name`*** [**`force`**]
Deletes the specified user in the local user list.

`#(config local-user-list` *`local_user_list`*`)` **`user edit`** ***`user_name`***
changes the prompt to `#(config local-user-list` *`local_user_list user_name`*`)`
Edits the specified user in the local user list.

`#(config local-user-list` *`local_user_list user_name`*`)` {**`disable`** | **`enable`**}
Disables/enables the user account.

`#(config local-user-list` *`local_user_list user_name`*`)` **`exit`**
Exits configure local-user-list *`user_list`* mode and returns to configure local-user-list mode.

`#(config local-user-list` *`local_user_list user_name`*`)` **`group`** {**`add`** | **`remove`**} ***`group_name`***
Adds/removes the specified group from the user.

`#(config local-user-list` *`local_user_list user_name`*`)` **`hashed-password`** *`hashed_password`*
Specifies the user's password in hashed format.

`#(config local-user-list` *`local_user_list user_name`*`)` **`password`** *`password`*
Specifies the user's password.

`#(config local-user-list` *`local_user_list user_name`*`)` **`view`**
Displays the user account.

`#(config local-user-list` *`local_user_list`*`)` **`view`**
Displays all users and groups in the local user list.

For More Information

- **`#(config) security local`** on page 339
- *SGOS Administration Guide*

Example

```
#(config) security local-user-list edit testlul
#(config local-user-list testlul) user create testuser
 ok
#(config local-user-list testlul) user edit testuser
#(config local-user-list testlul testuser) enable
 ok
#(config local-user-list testlul testuser) exit
#(config local-user-list testlul) exit
#(config)
```

#(config) security management

Synopsis

Manages the automatic logging out of a user and sets the name of realm in the Management Console challenge.

Syntax

```
#(config) security management [subcommands]
```

Subcommands

#(config) security management cli-timeout *minutes*

Specifies the length of an administrative CLI session before the administrator is required to re-enter credentials. The default is 15 minutes (900 seconds). Acceptable values are between 1 and 1440 minutes (60 seconds to 86400 seconds).

#(config) security management display-realm *realm_name*

Specifies the realm to display in the Management Console challenge. The default value is the IP address of the ProxySG appliance.

#(config) security management no cli-timeout

Disables the automatic session logout for CLI sessions.

#(config) **security management no display-realm**

Disables the specified web interface realm displayed in the Management Console challenge.

#(config) security management no web-timeout

Disables the automatic session logout for Management Console sessions.

#(config) security management web-timeout *minutes*

Specifies the length of an administrative Management Console session before the administrator is required to re-enter credentials. The default is 15 minutes (900 seconds). Acceptable values are between 1 and 1440 minutes (60 seconds to 86400 seconds).

For More Information

- *SGOS Administration Guide*

Example

```
#(config) security management web-timeout 20
  ok
```

#(config) security novell-sso

Synopsis

Allows you to configure and manage Novell SSO realms.

Syntax

```
#(config) security novell-sso [subcommands]
```

Subcommands

`#(config) security novell-sso create-realm realm_name`
Creates the specified Novell SSO realm.

`#(config) security novell-sso delete-realm realm_name`
Deletes the specified Novell SSO realm.

`#(config) security novell-sso edit-realm realm_name`
Changes the prompt. See Submodes for details.

`#(config) security novell-sso view [realm_name]`
Displays the configuration of all Novell SSO realms or just the configuration for *realm_name* if specified.

Submodes

`#(config) security novell-sso edit-realm realm_name`

This changes the prompt to:

`#(config novell-sso realm_name)`

Commands found in this submode include:

`SGOS#(config novell-sso realm_name) alternate-agent {host hostname | port port_number}`
Specifies the alternate agent hostname and port number.

`SGOS#(config novell-sso realm_name) alternate-agent private-key-password {private key password | <enter>}`
Sets the alternate private key password. Entering the submode without a password opens the password prompt followed by a confirmation prompt.

`SGOS#(config novell-sso realm_name) alternate-agent public-certificate-password {public certificate password | <enter>}`
Sets the alternate public certificate password. Entering the submode without a password opens the password prompt followed by a confirmation prompt.

`SGOS#(config novell-sso realm_name) alternate-agent encrypted-private-key-password {private key password | <enter>}`
Sets the alternate private key password.

`SGOS#(config novell-sso realm_name) alternate-agent encrypted-public-certificate-password {public certificate password | <enter>}`
Sets the alternate public certificate password.

`SGOS#(config novell-sso realm_name) authorization {realm-name authorization-realm-name | username username | no {authorization-realm-name | username} | self}`
Specifies the realm name, which can be **self**, and username for authorization. **No** clears the realm and username.

`SGOS#(config novell-sso` *realm_name*`)` **`cookie`** {**`persistent`** {**`disable`** | **`enable`**}| **`verify-ip`** {**`disable`** | **`enable`**}}
Specifies whether to enable persistent or session cookies, and whether to verify the IP address of the cookie.

`SGOS#(config novell-sso` *realm_name*`)` **`exit`**
Leaves the `novell-sso` edit-realm mode.

`SGOS#(config novell-sso` *realm_name*`)` **`full-search`** {**`day-of-week`** | **`time-of-day`**}
Specifies the day of the week for full searches to occurs and the time of the day (UTC time) to search.

`SGOS#(config novell-sso` *realm_name*`)` **`inactivity-timeout`** *seconds*
Specifies the amount of time a session can be inactive before being logged out.

`SGOS#(config novell-sso` *realm_name*`)` **`ldap monitor-server`** {**`add`** *LDAP_host* [*LDAP_port*]| **`clear`** | **`remove`** *LDAP_host* [*LDAP_port*]}
Add an LDAP host to list of servers to be monitored, clear the list, or remove a specific LDAP host from the list of servers to be monitored.

`SGOS#(config novell-sso` *realm_name*`)` **`ldap search-realm`** *ldap_realm*
Specifies the name of the realm to search and monitor.

`SGOS#(config novell-sso` *realm_name*`)` **`ldap-name`** {**`login-time`** *LDAP_name*| **`network-address`** *LDAP_name*}
Specifies the name of the LDAP server for Novell directory attributes.

`SGOS#(config novell-sso` *realm_name*`)` **`no alternate-agent`**
Removes the alternate agent.

`SGOS#(config novell-sso` *realm_name*`)` **`primary-agent`** {**`host`** *hostname* | **`port`** *port_number*}
Specifies the primary agent hostname and port number.

`SGOS#(config novell-sso` *realm_name*`)` **`refresh-time`** {**`authorization-refresh`** *seconds* | **`surrogate-refresh`** *seconds*}
Sets the refresh time for authorization and surrogates.

`SGOS#(config novell-sso` *realm_name*`)` **`rename`** *new_realm_name*
Renames the current realm to *new_realm_name.*

`SGOS#(config novell-sso` *realm_name*`)` **`ssl`** {**`enable`** | **`disable`**}
Enables or disables SSL between the ProxySG and the BCAAA service.

`SGOS#(config novell-sso` *realm_name*`)` **`ssl-device-profile`** *ssl_device_profile_name*
Specifies the device profile to use

`SGOS#(config novell-sso` *realm_name*`)` **`timeout`** *seconds*
The time allotted for each request attempt. The default is 60 seconds.

`SGOS#(config novell-sso` *realm_name*`)` **`test-authentication`** *IP_address*
Tests the Novell SSO and BCAAA configuration to ensure that the `ProxySG` appliance can successfully map an IP address to a user in your Novell Directory.

`SGOS#(config novell-sso` *realm_name*`)` **`view`**
Displays this realm's configuration.

`SGOS#(config novell-sso` *realm_name*`)` **`virtual-url`** *url*
Specifies the virtual URL to use for this realm. If no URL is specified the global transparent proxy virtual URL is used.

#(config) security password and hashed_password

Synopsis

Sets the console password to the password specified.

Syntax

```
#(config) security password password
#(config) security password hashed-password hashed_password
```

Subcommands

`#(config) security password password`
This is the password required to enter enable mode from the CLI when using console credentials, the serial console, or RSA SSH.

`#(config) security hashed-password hashed_password`
The password in hashed format. You can either hash the password prior to entering it, or you can allow the ProxySG to hash the password.

For More Information

- *SGOS Administration Guide*

Example

```
#(config) security password good2test
```

#(config) security password-display

Synopsis

Sets various display settings.

Syntax

```
#(config) security password-display [subcommands]
```

Subcommands

`#(config) security password-display {encrypted | none}`
Specifies the format to display passwords in `show config` output. Specify `encrypted` to display encrypted passwords. Specify `none` to display no passwords.

`#(config) security password-display keyring`
Specifies the keyring to use for password encryption.

`#(config) security password-display view`
Displays the current password display settings.

For More Information

- *SGOS Administration Guide*

Example

```
#(config) security password-display view
Password display mode: Encrypted
Password encryption keyring: configuration-passwords-key
```

#(config) security policy-substitution

A Policy Substitution realm provides a mechanism for identifying and authorizing users based on information in the request to the ProxySG. The realm uses information in the request and about the client to identify the user. The realm is configured to construct user identity information by using policy substitutions.

The Policy Substitution realm is used typically for best-effort user discovery, mainly for logging and subsequent reporting purposes, without the need to authenticate the user. Be aware that if you use Policy Substitution realms to provide granular policy on a user, it might not be very secure because the information used to identify the user can be forged.

Synopsis

Allows you to create and manage policy-substitution realms.

Syntax

`#(config) security policy-substitution [subcommands]`

Subcommands

`#(config) security policy-substitution create-realm realm_name`
Creates the specified policy-substitution realm

`#(config) security policy-substitution delete-realm realm_name`
Deletes the specified policy-substitution realm.

`#(config) security policy-substitution edit-realm realm_name`
Changes the prompt. See Submodes for details.

`#(config) security policy-substitution view [realm_name]`
Displays the configuration of all policy-substitution realms or just the configuration for *realm_name* if specified.

Submodes

`#(config) security policy-substitution edit-realm realm_name`

This changes the prompt to:

`#(config policy-substitution realm_name)`

Commands found in this submode include:

`#(config policy-substitution realm_name) authorization-realm-name realm_name`
This option is only required if you are associating an authorization realm with the Policy Substitution realm.

`#(config policy-substitution realm_name) cookie {persistent {disable | enable}| verify-ip {disable | enable}}`
Specifies whether to enable persistent or session cookies, and whether to verify the IP address of the cookie.

`#(config policy-substitution realm_name) exit`
Leaves the windows-sso edit-realm mode.

`#(config policy-substitution realm_name)group-definition substitution_string`
The authenticated group name for use in a policy substitution realm. This command identifies the string to be used in an HTTP header, for use in a child proxy in proxy chain configurations. The parent proxy would look for the HTTP header string and through policy actions, make policy decisions based on a user's group.

Note: This command has no impact when used in a policy substitution realm that includes an authorization realm. For more information on Policy Substitution realms, refer to the *Blue Coat SGOS 6.5 Policy Language Reference.*

`#(config policy-substitution realm_name) identification determine-usernames {by-definition cr | by-search cr}`
Specifies how to determine usernames.

`#(config policy-substitution realm_name) identification full-username construction_rule`
The full username as created through policy substitutions. The construction rule is made up any of the substitutions whose values are available at client logon, listed in Appendix D, "CPL Substitutions," in the *Blue Coat Content Policy Language Reference*.

Note: The username and full username attributes are character strings that contain policy substitutions. When authentication is required for the transaction, these character strings are processed by the policy substitution mechanism, using the current transaction as input. The resulting string is stored in the user object in the transaction, and becomes the user's identity.

To create full usernames for various uses in Policy Substitution realms, refer to the *Blue Coat SGOS 6.5 Content Policy Language Reference*.

`#(config policy-substitution realm_name) identification ignore-user-list {add username | clear cr | remove username}`
Specifies users to ignore when determining usernames by search.

`#(config policy-substitution realm_name) identification realm-name LDAP realm`
Specifies the name of the LDAP search realm.

`#(config policy-substitution realm_name) identification search-filter search filter`
Specifies the LDAP search filter.

`#(config policy-substitution realm_name) identification username construction_rule`
The username as created through policy substitutions. The username is only required if you are using an authorization realm. The construction rule is made up any of the policy substitutions whose values are available at client logon, listed in Appendix D, "CPL Substitutions," in the *Blue Coat SGOS 6.5 Content Policy Language Reference*.

Note: The username and full username attributes are character strings that contain policy substitutions. When authentication is required for the transaction, these character strings are processed by the policy substitution mechanism, using the current transaction as input. The resulting string is stored in the user object in the transaction, and becomes the user's identity.

To create usernames for the various uses of Policy Substitution realms, refer to the *SGOS 6.5 Content Policy Language Reference*.

`#(config policy-substitution realm_name) inactivity-timeout seconds`
Specifies the amount of time a session can be inactive before being logged out.

`#(config policy-substitution realm_name) no authorization-realm-name`
Clears the authorization realm name.

`#(config policy-substitution realm_name) refresh-time {authorization-refresh seconds | surrogate-refresh seconds}`
Sets the refresh time for authorization and surrogates.

`#(config policy-substitution` *`realm_name`*`)` **`rename`** *`new_realm_name`*
Renames this realm to *`new_realm_name`*.

`#(config policy-substitution` *`realm_name`*`)` **`view`**
Displays this realm's configuration.

`#(config policy-substitution` *`realm_name`*`)` **`virtual-url`** *`url`*
Specifies the virtual URL to use for this realm. If no URL is specified the global transparent proxy virtual URL is used.

For More Information

- *SGOS Administration Guide,* Access Logging chapter
- *SGOS 6.x Visual Policy Manager Reference*

Example

```
#(config) security policy-substitution edit-realm PS1
#(config policy-substitution PS1) authorization-realm-name LDAP1
#(config policy-substitution PS1) username $(netbios.messenger-username)
#(config policy-substitution PS1) identification full-username
cn=$(netbios.messenger-username),cn=users,dc=$(netbios.computer-domain),
dc=company,dc=com
```

#(config) security radius

RADIUS is often the protocol of choice for ISPs or enterprises with very large numbers of users. RADIUS is designed to handle these large numbers through centralized user administration that eases the repetitive tasks of adding and deleting users and their authentication information. RADIUS also inherently provides some protection against sniffing.

Some RADIUS servers support one-time passwords. One-time passwords are passwords that become invalid as soon as they are used. The passwords are often generated by a token or program, although pre-printed lists are also used. Using one-time passwords ensures that the password cannot be used in a replay attack.

The ProxySG appliance's one-time password support works with products such as Secure Computing SafeWord synchronous and asynchronous tokens and RSA SecurID tokens.

The ProxySG supports RADIUS servers that use challenge/response as part of the authentication process. SafeWord asynchronous tokens use challenge/response to provide authentication. SecurID tokens use challenge/response to initialize or change PINs.

Synopsis

Allows you to create and manage RADIUS realms.

Syntax

```
#(config) security radius [subcommands]
```

Subcommands

```
#(config) security radius create-realm realm_name
```
Creates the specified RADIUS realm

```
#(config) security radius create-realm-encrypted <realm_name> <encrypted_secret>
   <primary-server_host> [<primary-server port>]
```
Creates a RADIUS realm with an encrypted server secret.

```
#(config) security radius create-realm-prompt-secret <realm_name>
   <primary-server_host> [<primary-server port>]
```
Creates the specified RADIUS realm; prompts for a server secret.

```
#(config) security radius delete-realm realm_name
```
Deletes the specified RADIUS realm.

```
#(config) security radius edit-realm realm_name
```
Changes the prompt. See Submodes for details.

```
#(config) security radius view [realm_name]
```
Displays the configuration of all RADIUS realms or just the configuration for *realm_name* if specified.

Submodes

```
#(config) security radius attributes
```

This changes the prompt to:

```
#(config radius attributes)
```

Commands found in this submode include:

```
#(config radius attributes) add {radius-attribute <radius-type (1-255)>
   <attribute name> [integer|ipv4|ipv6]|[string <max-length (1-253)>] | vendor
   attribute <vendor id> <vendor-type (1-255)> <attribute name>
   [integer|ipv4|ipv6]|[string <max-length (1-247)>]}
```
Enables the user to specify the configuration of the RADIUS or vendor-specific attribute.

`#(config radius attributes)` **`exit`**
Return to the `#(config)` prompt.

`#(config radius attributes)` **`remove`** *`attribute_name`*
Removes the specified RADIUS attribute.

`#(config radius attributes)` **`view`**
View the configured RADIUS attributes.

`#(config)` **`security radius edit-realm`** *`realm_name`*

This changes the prompt to:

`#(config radius` *`realm_name`*`)`

Commands found in this submode include:

`#(config radius` *`realm_name`*`)` **`alternate-server encrypted-secret`** *`encrypted_secret`*
Specifies the alternate server secret in encrypted format. Note that you must create the encrypted secret before executing the `host [`*`port`*`]` command.

`#(config radius` *`realm_name`*`)` **`alternate-server`** *`host`* `[`*`port`*`]`
Specifies the alternate server host and port.

`#(config radius` *`realm_name`*`)` **`alternate-server secret`** *`secret`*
Specifies the alternate server secret. Note that you must create the secret before executing the `host [`*`port`*`]` command

`#(config radius` *`realm_name`*`)` **`case-sensitive`** {**`disable | enable`**}
Specifies whether or not the RADIUS server is case-sensitive.

`#(config radius` *`realm_name`*`)` **`cookie`** {**`persistent`** {**`enable`** | **`disable`**} | **`verify-ip`** {**`enable | disable`**}
Specifies whether to enable persistent or session cookies, and whether to verify the IP address of the cookie.

`#(config radius` *`realm_name`*`)` **`display-name`** *`display_name`*
Specifies the display name for this realm.

`#(config radius` *`realm_name`*`)` **`exit`**
Exits configure radius-realm mode and returns to configure mode.

`#(config radius` *`realm_name`*`)` **`inactivity-timeout`** *`seconds`*
Specifies the amount of time a session can be inactive before being logged out.

`#(config radius` *`realm_name`*`)` **`log-out`** {**`challenge`** {**`enable | disable`**} | **`display-time`** *`seconds`*}
Allows you to challenge the user after log out and define the log out page display time.

`#(config radius` *`realm_name`*`)` **`no alternate-server`**
Clears the alternate-server.

`#(config radius` *`realm_name`*`)` **`one-time-passwords`** {**`enable | disable`**}
Allows you to use one-time passwords for authentication. The default is disabled.

`#(config radius` *`realm_name`*`)` **`primary-server encrypted-secret`** *`encrypted_secret`*
Specifies the primary server secret in encrypted format.

`#(config radius` *`realm_name`*`)` **`primary-server`** *`host`* `[`*`port`*`]`
Specifies the primary server host and port.

`#(config radius` *`realm_name`*`)` **`primary-server secret`** *`secret`*
Specifies the primary server secret.

`#(config radius` *`realm_name`*`)` **`refresh-time`** {**`credential-refresh`** *`seconds`* | **`rejected-credentials-refresh`** *`seconds`* | **`surrogate-refresh`** *`seconds`*}
Sets the refresh time for credential, rejected credentials cache, and surrogates.

`#(config radius realm_name) rename new_realm_name`
Renames this realm to *new_realm_name*.

`#(config radius realm_name) server-retry count`
Specifies the number of authentication retry attempts. This is the number of attempts permitted before marking a server offline. The client maintains an average response time from the server; the retry interval is initially twice the average. If that retry packet fails, then the next packet waits twice as long again. This increases until it reaches the timeout value. The default number of retries is 10.

`#(config radius realm_name) server-authentication {none | origin | proxy}`
Enables/disables the forwarding of BASIC credentials of the authenticated user to the origin content server or for proxy authentication. Flush the entries for a realm if the server-authentication value is changed to ensure that the server-authentication value is immediately applied.

You can only choose one server-authentication method:

- If set to **origin**, BASIC credentials are forwarded to an upstream server.
- If set to **proxy**, BASIC credentials are forwarded to an upstream proxy.
- If set to **none**, forwarding BASIC credentials is disabled.

`#(config radius realm_name) test-authentication username password`
Tests the RADIUS configuration to ensure that the ProxySG can successfully authenticate a user in your RADIUS realm. If the test succeeds, the CLI displays a list of groups to which the user belongs.

`#(config radius realm_name) timeout seconds`
Specifies the RADIUS request timeout. This is the number of seconds the ProxySG allows for each request attempt before giving up on a server and trying another server. Within a timeout multiple packets can be sent to the server, in case the network is busy and packets are lost. The default request timeout is 10 seconds.

`#(config radius realm_name) server-charset charset`
Allows you to select the character set you need. A character set is a MIME charset name. Any of the standard charset names for encodings commonly supported by Web browsers can be used. The default is Unicode:UTF8.

One list of standard charset names is found at http://www.iana.org/assignments/character-sets.

`#(config radius realm_name) view`
Displays this realm's configuration.

`#(config radius realm_name) virtual-url` *url*
Specifies the virtual URL to use for this realm. If no URL is specified the global transparent proxy virtual URL is used.

For More Information

- *SGOS Administration Guide*

Example

```
#(config) security radius edit-realm testradius
#(config radius testradius) server-retry 8
 ok
#(config radius testradius) server-authentication proxy
 ok
#(config radius testradius) exit
```

#(config) security request-storage

When a request requiring the user to be challenged with a form contains a body, the request is stored on the ProxySG while the user is being authenticated. Storage options include:

- the maximum request size.
- the expiration of the request.
- whether to verify the IP address of the client requesting against the original request.
- whether to allow redirects from the origin server

The storage options are global, applying to all form exceptions you use.

The global allow redirects configuration option can be overridden on a finer granularity in policy using the `authenticate.redirect_stored_requests(yes|no)` action.

Synopsis

Used with authentication forms to store requests.

Syntax

```
#(config) security request-management [subcommands]
```

Subcommands

`#(config) security request-management allow-redirects {disable | enable}`
Specifies whether to allow redirects. The default is `disable`.

`#(config) security request-management expiry-time seconds`
Sets the amount of time before the stored request expires. The default is `300` seconds (five minutes).

`#(config) security request-management max-size megabytes`
Sets the maximum POST request size during authentication. The default is `50` megabytes.

`#(config) security request-management verify-ip {disable | enable}`
Enables or disables the verify-ip option. The default is to enable the ProxySG to verify the IP address against the original request.

For More Information

- **#(config) security authentication-forms** on page 314
- *SGOS Administration Guide*

Example

```
#(config) security request-storage max-size megabytes
#(config) security request-storage expiry-time seconds
#(config) security request-storage verify-ip enable | disable
#(config) security request-storage allow-redirects enable | disable
```

#(config security saml)

SAML 2.0 was developed by the OASIS Security Services Technical Committee. It is an industry standard for retrieving authorization and identity information in XML documents to facilitate single sign-on (SSO) applications or services on the internet. In SAML authentication, the exchange of information is performed by the following entities:

Identity providers (IDPs), which are identity stores. For example, an IDP may have a back-end directory of users. The IDP authenticates the users. Supported IDPs are:

- Microsoft® Active Directory Federation Services (AD FS) 2.0
  Note: ADFS 1.0 ships with Windows Server 2008. If you want to use the SAML realm with AD FS, you must download AD FS 2.0 from the Microsoft website and install it.
- CA SiteMinder® R12
- Oracle® Access Management 11g
- (Available in SGOS 6.5.2 and later) Shibboleth 2.3.5

Service providers (SPs), which provide access to applications or services to users. It is the entity against which users authenticate. SGOS supports SAML authentication in which the ProxySG acts as the SP.

Synopsis

Allows you to create and configure SAML realms.

Syntax

`#(config)security saml [subcommands]`

Subcommands

`#(config)security saml create-realm <realm-name>`

Creates a realm.

`#(config)security saml view <realm-name>`

Shows running system information for all SAML realms or the specified SAML realm.

`#(config)security saml delete-realm <realm-name>`

Deletes the specified SAML realm.

`#(config)security saml edit-realm <realm-name>`

Changes the prompt. See Submodes for details.

Submodes

`#(config)security saml attributes`

Configures SAML attributes. This changes the prompt to:

`#(config saml attributes)`

Commands in this submode include:

`#(config saml attributes)add <attribute-name>`

Adds a new SAML attribute.

`#(config saml attributes)`**edit <*attribute-name*>**

Edits the specified SAML attribute.

`#(config saml attributes)`**remove <*attribute-name*>**

Removes the specified SAML attribute.

`#(config saml attributes)`**view**

Displays the configured SAML attributes.

`#(config)`**security saml edit-realm <*realm-name*>**

Edits the realm. This changes the prompt to:

`#(config saml <realm-name>)`

Commands in this submode include:

`#(config saml <realm-name>)`**authorization ignore-user-list {add | clear | remove}**

Add a username to a list of users to ignore when determining authorization, clear the list, or remove a username from the list.

`#(config saml <realm-name>)`**authorization realm {none | <*realm-name*> | self}**

Specify whether to not authorize with the current realm, use a different realm for authorization, or authorize with the current realm.

`#(config saml <realm-name>)`**authorization search-filter <*search-filter*>**

Specify the LDAP search filter.

`#(config saml <realm-name>)`**authorization search-realm <*LDAP-realm*>**

Specify the name of the LDAP search realm.

`#(config saml <realm-name>)`**authorization user-attribute { fqdn | <*LDAP-attribute-name*>}**

Specify the username attribute on the search result object— either the FQDN or the LDAP attribute name.

`#(config saml <realm-name>)`**authorization username {determine-by-search | use-full-username | <*username*>}**

Set the username for authorization: Determine the username by LDAP search, use the user's full username or FQDN, or specify the username.

`#(config saml <realm-name>)`**client-redirects {disable | enable}**

(Available in SGOS 6.5.2 and later) Specify if SAML redirects should be forwarded to the client.

`#(config saml <realm-name>)`**cookie persistent {disable | enable}**

Specify whether to use persistent or session cookies.

`#(config saml <realm-name>)`**cookie verify-ip {disable | enable}**

Specify whether to verify cookies' IP addresses.

`#(config saml <realm-name>)`**display-name <*display-name*>**

Set the display name of the current realm.

`#(config saml <realm-name>)`**encryption keyring <*keyring-name*>**

Specify the keyring used for decrypting assertions.

`#(config saml <realm-name>)federated-idp {ccl <ccl-name> | entity-id <entity-id>| import-metadata <url> | sso-post-endpoint <url>| sso-redirect-endpoint <url>}`

Configure the following settings for the IDP with which the realm is federated:

- the trusted CCL for validation of the IDP certificate
- the SAML entity ID
- the URL from which the IDP metadata is downloaded/imported; the URL is not stored, but it is used to import metadata when the command is issued
- the SSO POST endpoint
- the SSO redirect endpoint

`#(config saml <realm-name>)group-attribute <attribute-name>`

Specify the name of the group membership attribute.

`#(config saml <realm-name>)inactivity-timeout <number-of-seconds>`

Specify the number of seconds a session can be inactive before it is logged out.

`#(config saml <realm-name>)inline idp-metadata <XML> <EOF>`

Install IDP metadata by entering it in XML format, followed by an end-of-file character.

`#(config saml <realm-name>)log-out challenge {disable | enable}`

Disable or enable challenging after logout. For example, if this setting is enabled and a user logs out of a web site, the user must enter credentials again the next time they access the web site.

`#(config saml <realm-name>)log-out display-time <number-of-seconds>`

Specify the number of seconds to display the logout page after logging out.

```
#(config saml <realm-name>)no {encryption keyring | federated-idp
   sso-post-endpoint | federated-idp sso-redirect-endpoint |
   group-attribute | user-attribute | user-fullname-attribute }
```

Clear the specified parameter.

`#(config saml <realm-name>)not-after <number-of-seconds>`

Specify a number of seconds after the current time, after which assertions are invalid. The default value is 20.

`#(config saml <realm-name>)not-before <number-of-seconds>`

Specify a number of seconds before the current time, before which assertions are invalid. The default value is 10.

`#(config saml <realm-name>)prefix-idp-cookies {disable | enable}`

(Available in SGOS 6.5.2 and later) Specify if IDP cookies should be prefixed when client redirect is disabled.

```
#(config saml <realm-name>)refresh-time { credential-refresh
   <number-of-seconds> | rejected-credential-refresh <number-of-seconds>
   | surrogate-refresh <number-of-seconds> }
```

Configure the refresh time for authorization credentials and surrogates.

`#(config saml <realm-name>)rename <new-realm-name>`

Rename the current realm.

`#(config saml <realm-name>)require-encryption {disable | enable}`

Disable or enable the requirement that all incoming assertions are encrypted.

`#(config saml <realm-name>)ssl-device-profile <ssl-device-profile>`

(Available in SGOS 6.5.2 and later) Specify the SSL device profile to use for the realm.

`#(config saml <realm-name>)user-attribute <attribute-name>`

Specify the attribute that contains the username.

`#(config saml <realm-name>)user-fullname-attribute <attribute-name>`

Specify the attribute that contains the full username.

`#(config saml <realm-name>)view`

Show running system information for the current SAML realm.

`#(config saml <realm-name>)virtual-host <hostname>`

Specify the hostname for the SAML endpoints.

`#(config)security saml attributes`

This changes the prompt to:

```
#(config saml attributes)
```

`#(config saml attributes)edit <attribute-name>`

This changes the prompt to:

```
#(config saml attributes <attribute-name>)
```

`#(config saml attributes <attribute-name>)data type {case-ignore-string | case-exact-string}`

Change the specified attribute's data type.

`#(config saml attributes <attribute-name>)saml-name <saml-name>`

Change the specified attribute's SAML name.

Example

The following example shows output in SGOS 6.5.3.

```
#(config)security saml view realm1
  Realm name:                 realm1
  Display name:               realm1
  Federated IDP entity ID:
  Federated IDP POST URL:
  Federated IDP Redirect URL:
  Federated IDP CCL:          appliance-ccl
  SSL Device Profile Name:    default
  Not Before:                 60
  Not After:                  60
  SAML user attribute:
  SAML fullname attribute:
```

```
SAML group attribute:
SAML encryption keyring:
Require encryption:          no
Authorization realm:         Self
Authorization username:      Full Authorization Username
Search realm:
Search filter:
User attribute:              Entry FQDN
Users to ignore:
Virtual URL:                 www.cfauth.com/
Credentials refresh:         900
Surrogates refresh:          900
Inactivity timeout:          900
Verify cookie ip address:    yes
Use persistent cookies:      no
Challenge after log out:     yes
Log out page display time:   0
Rejected credentials time:   1
```

#(config security sequence)

After a realm is configured, you can associate it with other realms to allow Blue Coat to search for the proper authentication credentials for a specific user. That is, if the credentials are not acceptable to the first realm, they are sent to the second, and so on until a match is found or all the realms are exhausted. This is called *sequencing*.

Synopsis

Allows you to create and manage sequence realms.

Syntax

`#(config) security sequence [subcommands]`

Subcommands

`#(config) security sequence create-realm realm_name`
Creates the specified sequence realm

`#(config) security sequence delete-realm realm_name`
Deletes the specified sequence realm.

`#(config) security sequence edit-realm realm_name`
Changes the prompt. See Submodes for details.

`#(config) security sequence view [realm_name]`
Displays the configuration of all sequence realms or just the configuration for *realm_name* if specified.

`#(config) security sequence edit-realm realm_sequence_name`

This changes the prompt to:

`#(config sequence realm_sequence_name)`

Submodes

Commands available in this submode include:

`#(config sequence realm_sequence_name) display-name display_name`
Specifies the display name for this realm.

`#(config sequence realm_sequence_name) exit`
Exits configure sequence-realm mode and returns to configure mode.

`#(config sequence realm_sequence_name) IWA-only-once {disable | enable}`
Specifies whether or not to challenge for credentials for the IWA realm one or multiple times.

`#(config sequence realm_sequence_name) realm {add | demote | promote | remove} {realm_name | clear}`
Adds/demotes/promotes/removes a realm from the realm sequence, or clears all realms from the realm sequence.

`#(config sequence realm_sequence_name) rename new_realm_name`
Renames this realm to *new_realm_sequence_name*.

`#(config sequence realm_sequence_name) try-next-realm-on-error {disable | enable}`
Use this command to specify that the next realm on the list should be attempted if authentication in the previous realm has failed with a permitted error. The default value is to not attempt the next realm and fall out of the sequence.

`#(config sequence realm_sequence_name) view`
Displays this realm's configuration.

`#(config sequence realm_sequence_name) virtual-url url`
Specifies the virtual URL to use for this realm sequence. If no URL is specified the global transparent proxy virtual URL is used.

For More Information

- *SGOS Administration Guide*

Example

```
#(config) security sequence edit-realm testsequence
#(config sequence testsequence) IWA-only-once disable
 ok
#(config sequence testsequence) realm clear
 ok
#(config sequence testsequence) exit
```

#(config) security siteminder

Within the SiteMinder system, BCAAA acts as a custom Web agent. It communicates with the SiteMinder policy server to authenticate the user and to obtain a SiteMinder session token, response attribute information, and group membership information.

Custom header and cookie response attributes associated with **OnAuthAccept** and **OnAccessAccept** attributes are obtained from the policy server and forwarded to the ProxySG. They can (as an option) be included in requests forwarded by the *appliance*.

Within the ProxySG system, BCAAA acts as its agent to communicate with the SiteMinder server. The ProxySG provides the user information to be validated to BCAAA, and receives the session token and other information from BCAAA.

Each ProxySG SiteMinder realm used causes the creation of a BCAAA process on the Windows host computer running BCAAA. A single host computer can support multiple ProxySG realms (from the same or different ProxySG appliances); the number depends on the capacity of the BCAAA host computer and the amount of activity in the realms.

Note: Each (active) SiteMinder realm on the ProxySG should reference a different agent on the Policy Server.

Configuration of the ProxySG's realm must be coordinated with configuration of the SiteMinder policy server. Each must be configured to be aware of the other. In addition, certain SiteMinder responses must be configured so that BCAAA gets the information the ProxySG needs.

Synopsis

Allows you to create and manage SiteMinder realms.

Syntax

```
#(config) security siteminder [subcommands]
```

Subcommands

`#(config) security siteminder create-realm realm_name`
Creates the specified SiteMinder realm

`#(config) security siteminder delete-realm realm_name`
Deletes the specified SiteMinder realm.

`#(config) security siteminder edit-realm realm_name`
Changes the prompt. See Submodes for details.

`#(config) security siteminder view [realm_name]`
Displays the configuration of all SiteMinder realms or just the configuration for *realm_name* if specified.

Submodes

```
#(config) security siteminder edit-realm realm_name
```

This changes the prompt to:

```
#(config siteminder realm_name)
```

Commands in this submode include:

`#(config siteminder realm_name) add-header-responses {enable | disable}`
Enable if your Web applications need information from the SiteMinder policy server responses.

`#(config siteminder realm_name) alternate-agent agent-name agent_name`
Specifies the alternate agent.

`#(config siteminder realm_name) alternate-agent encrypted-shared-secret encrypted-shared-secret`
Specifies the alternate agent secret in encrypted format.

`#(config siteminder realm_name) alternate-agent host host`
The host ID or the IP address of the system that contains the alternate agent.

`#(config siteminder realm_name) alternate-agent port port`
The port where the agent listens.

`#(config siteminder realm_name) alternate-agent shared-secret secret`
Specifies the alternate agent secret.

`#(config siteminder realm_name) alternate-agent always-redirect-offbox`
Enables or disables SSO.

`#(config certificate realm_name) authorization {ignore-user-list {add | clear | remove}`
Manages the ignore-user-list, which is the list of those to ignore if they are returned as search results.

`##(config siteminder realm_name) authorization realm {none | realm-name realm_name}`
Specifies the authorization realm to use. Only LDAP, XML, and local realms are valid authorization realms.

`#(config siteminder realm_name) authorization search-filter search_filter`
Specifies the search filter that should be used during a search of the LDAP server. The filter can contain policy substitutions including `$(cs-username)`.

`#(config siteminder realm_name) authorization search-realm LDAP_realm`
Specifies the name of the LDAP search realm.

`#(config siteminder realm_name) authorization user-attribute {fqdn | LDAP_attribute_name}`
Specifies the user-attribute (fully qualified domain name or an LDAP attribute name) to be used during a search of the LDAP server.

`#(config siteminder realm_name) authorization username {determine-by-search | use-full-username | username_for_authorization}`
Specifies the way a username should be determined. The default is the attribute `cn`, which specifies the user's relative name.

`#(config siteminder realm_name) always-redirect-offbox {enable | disable}`
The ProxySG realm can be configured to redirect to an off-box authentication service always. The URL of the service is configured in the scheme definition on the SiteMinder policy server. The ProxySG realm is then configured with `always-redirect-offbox` enabled.

`#(config siteminder realm_name) case-sensitive {enable | disable}`
Specifies whether the SiteMinder server is case-sensitive.

`#(config siteminder realm_name) cookie {persistent {enable | disable} | verify-ip {enable | disable}`

Specifies whether to enable persistent or session cookies, and whether to verify the IP address of the cookie.

`#(config siteminder realm_name) display-name display_name`
Specifies the display name for this realm.

`#(config siteminder realm_name) exit`
Exits configure siteminder-realm mode and returns to configure mode.

`#(config siteminder realm_name) inactivity-timeout seconds`
Specifies the amount of time a session can be inactive before being logged out.

`#(config siteminder realm_name) log-out {challenge {enable | disable} | display-time seconds}`
Allows you to challenge the user after log out and define the log out page display time.

`#(config siteminder realm_name) no alternate-agent`
Clears the alternate agent configuration.

`#(config siteminder realm_name) primary-agent agent name agent_name`
Specifies the primary agent.

`#(config siteminder realm_name) primary-agent encrypted-shared-secret encrypted-shared-secret`
Specifies the primary agent secret in encrypted format.

`#(config siteminder realm_name) primary-agent host host`
The host ID or the IP address of the system that contains the primary agent.

`#(config siteminder realm_name) primary-agent port port`
The port where the agent listens.

`#(config siteminder realm_name) primary-agent shared-secret secret`
Specifies the primary agent secret.

`#(config siteminder realm_name) protected-resource-name resource-name`
The protected resource name is the same as the resource name on the SiteMinder server that has rules and policy defined for it.

`#(config siteminder realm_name) refresh-time {credential-refresh seconds | rejected-credentials-refresh seconds | surrogate-refresh seconds}`
Sets the refresh time for credential, rejected credentials cache, and surrogates.

`#(config siteminder realm_name) rename new_realm_name`
Renames this realm to *new_realm_name.*

`#(config siteminder realm_name) server-mode {failover | round-robin}`
Behavior of the server. Failover mode falls back to one of the other servers if the primary one is down. Round-robin modes specifies that all of the servers should be used together in a round-robin approach. Failover is the default

`#(config siteminder realm_name) siteminder-server create server_name`
Creates a SiteMinder server.

`#(config siteminder realm_name) siteminder-server delete server_name`
Deletes a SiteMinder server.

`#(config siteminder realm_name) siteminder-server edit server_name`
This changes the prompt to `#(config siteminder realm_name server_name).`

`#(config siteminder realm_name server_name) accounting-port port_number`
The default is 44441. The ports should be the same as the ports configured on the SiteMinder policy server. The valid port range is 1-65535.

`#(config siteminder realm_name server_name) authentication-port port_number`
The default is 44442. The ports should be the same as the ports configured on the SiteMinder server. The valid port range is 1-65535.

`#(config siteminder realm_name server_name)` **`authorization-port`** *`port_number`*
The default is 44443. The ports should be the same as the ports configured on the SiteMinder server. The valid port range is 1-65535.

`#(config siteminder realm_name server_name)` **`connection-increment`** *`number`*
The default is 1. The connection increment specifies how many connections to open at a time if more are needed and the maximum is not exceeded.

`#(config siteminder realm_name server_name)` **`exit`**
Leaves the `server_name` prompt and returns to the SiteMinder *`realm_name`* prompt.

`#(config siteminder realm_name server_name)` **`ip-address`** *`ip_address`*
The IP address of the SiteMinder server.

`#(config siteminder realm_name server_name)` **`max-connections`** *`number`*
The default is 256. The maximum number of connections is 32768.

`#(config siteminder realm_name server_name)` **`min-connections`** *`number`*
The default is 1.

`#(config siteminder realm_name server_name)` **`timeout`** *`seconds`*
The default is 60.

`#(config siteminder realm_name server_name)` **`view`**
Displays the server's configuration.

`#(config siteminder realm_name)` **`ssl {enable | disable}`**
Disables/enables SSL communication between the ProxySG and BCAAA.

`#(config siteminder realm_name)` **`ssl-device-profile`** *`ssl_device_profile_name`*
Specifies the device profile to use.

`#(config siteminder realm_name)` **`timeout`** *`seconds`*

`#(config siteminder realm_name)` **`validate-client-ip {disable | enable}`**
Enables validation of the client IP address. If the client IP address in the SSO cookie might be valid yet different from the current request client IP address, due to downstream proxies or other devices, disable client IP validation. The SiteMinder agents participating in SSO with the ProxySG should also be modified. The TransientIPCheck variable should be set to `yes` to enable IP validation and `no` to disable it.

Enable is the default.

`#(config siteminder realm_name)` **`view`**
Displays this realm's configuration.

`#(config siteminder realm_name)` **`virtual-url`** *`url`*
Specifies the virtual URL to use for this SiteMinder realm. If no URL is specified the global transparent proxy virtual URL is used.

For More Information

- *SGOS Administration Guide*

Example

```
#(config) security siteminder edit-realm test2
#(config siteminder test2) server-mode round-robin
 ok
#(config siteminder test2) ssl enable
 ok
#(config siteminder test2) exit
```

#(config) security transparent-proxy-auth

Synopsis

Configures authentication method for transparent proxies.

Syntax

```
#(config) security transparent-proxy-auth [subcommands]
```

Subcommands

```
#(config) security transparent-proxy-auth method {ip | cookie}
```

Specifies whether to use IP or cookie surrogate credentials.

```
#(config) security transparent-proxy-auth meta-refresh {enable | disable}
```

Enables or disables meta-refresh style redirects with Internet Explorer. Some browsers, such as Internet Explorer, have a hard-coded limit on the number of server redirects that they follow for a given request. For example, if you are browsing a website that performs several redirects, the redirects added by the ProxySG appliance authentication subsystem can exceed the browser's limit. The end result is that the browser will refuse to load the webpage, because it will not follow all of the redirects. You can use the `security transparent-proxy-auth meta-refresh enable` command to allow the ProxySG appliance to perform redirects for authentication without exceeding the browser's limit. When this CLI setting is enabled, the ProxySG appliance redirects the browser by sending an HTTP 200 response with a meta-refresh header, rather than by sending an HTTP 302 or 307 response.

Examples

```
#(config) security transparent-proxy-auth method cookie
#(config) security transparent-proxy-auth meta-refresh enable
```

#(config) security trust-package

Synopsis

Configures the settings for trust package updates. The trust package contains updates to the CA Certificate Lists (CCLs) and their associated CA certificates for the `browser-trusted` and `image-validation` CCLs.

Syntax

`#(config) **security trust-package** [*subcommands*]`

Subcommands

`#(config) **security trust-package download-path** *url*`
Specifies the URL from which the ProxySG appliance should download trust package updates. By default, the URL is set to `http://appliance.bluecoat.com/sgos/trust_package.bctp`. If you want to host your own download site, you can change the `download-path` to an on-site URL. In this case you must manually post the `trust_package.bctp` from the Blue Coat website on your download server; the ProxySG appliance can only download and install trust packages created by Blue Coat Systems, Inc.

`#(config) **security trust-package auto-update** {**enable** | **disable** | **interval** *days*}`
Enables or disables automatic trust package updates and/or sets the update interval. By default, automatic updates are enabled and have an interval of seven days. The interval can range from 1-30 days

For More Information

- *SGOS Administration Guide*

Example

```
#(config) security trust-package download-path
     http://download.acme.com/bluecoat/trust_package.bctp
#(config) security trust-package auto-update interval 10
```

#(config) security users

Synopsis

Allows administrators to manage user log ins, logouts and refresh data.

Syntax

```
#(config) security users
```

This changes the prompt to:

```
#(config users) [subcommands]
```

Subcommands

`#(config users) authorization-refresh {ip-addresses prefix [realm_name] | realms [realm_name] | users glob_user_name [realm_name]}`
Refreshes authorization data for the specified IP address, realm (or all realms), or user.

The IP address subnet notation is based on Classless Inter-Domain_Routing (CIDR):

- 1.2.3.4 : the IP address 1.2.3.4
- 1.2.3.0/24: the subnet 1.2.3.0 with netmask 255.255.255.0

The username pattern is a glob-based pattern, supporting three operators:

- '*' : match zero or more characters
- '?' : match exactly one character
- '[x-y]': match any character in the character range from 'x' to 'y'

`#(config users) credentials-refresh {ip-addresses prefix [realm_name] | realms [realm_name] | users glob_user_name [realm_name]}`
Refreshes credential data for the specified IP address, realm (or all realms), or user.

`#(config users) exit`
Returns to the `#(config)` prompt.

`#(config users) log-out {ip-addresses prefix [realm_name] | realms [realm_name] | users glob_user_name [realm_name]}`
Logs out the specified IP address, realm (or all realms), or user.

`#(config users) surrogates-refresh {ip-addresses prefix [realm_name] | realms [realm_name] | users glob_user_name [realm_name]}`
Refreshes surrogate data for the specified IP address, realm (or all realms), or user.

`#(config users) view detailed {ip-addresses prefix [realm_name] | realms [realm_name] | users glob_user_name [realm_name]}`
See a detailed view of users, sorted by IP address, realm, or username.

`#(config users) view {ip-addresses prefix [realm_name] | realms [realm_name] | users glob_user_name [realm_name]}`
See all logged-in users sorted by IP address, realm, or username.

For More Information

- *SGOS Administration Guide*

Example

```
#(config) security users
#(config users) surrogates-refresh ip-addresses 10.25.36.0/24
```

#(config) security username

Synopsis

Sets the console username.

Syntax

```
#(config) security username name
```

For More Information

- *SGOS Administration Guide*

Example

```
#(config) security username QATest
```

#(config security windows-domains)

Configures a Windows domain for features that require the appliance to join a domain, such as encrypted MAPI and IWA Direct.

Synopsis

First, you must create a hostname for the ProxySG appliance; the hostname you create must be unique within your Active Directory. You will not be able to join any domains until you have created a hostname. And, after you have joined a domain, you will not be able to modify this hostname. To do so, you would have to leave all domains you have joined and then rejoin them after you save a new hostname.

After you create the domain, you can join one or more Windows domains. To join a domain, you must first create a domain name alias and then you can join the domain using this alias.

Syntax

```
#(config) security windows-domains [subcommands]
```

Subcommands

```
#(config security windows-domains) create domain_name_alias

#(config security windows-domains) delete domain_name_alias

#(config security windows-domains)hostname ProxySG_hostname

#(config security windows-domains) inline domain-details domain_name_alias

#(config security windows-domains) join domain_name_alias DNS_domain_name
   join_account_name [join_account_password]

#(config security windows-domains) leave host domain_name [join_account_name
   [join_account_password]]

#(config security windows-domains) rejoin domain_name_alias join_account_name
   [join_account_password]

#(config security windows-domains) view
```

For More Information

- *SGOS Administration Guide*

Example

```
#(config) security windows-domains

#(config security windows-domains) hostname HQ1

#(config security windows-domains) create cal

#(config security windows-domains) join cal corp.example.com administrator
   testpass
```

#(config) security windows-sso

In a Windows SSO realm, the client is never challenged for authentication. Instead, the BCAAA agent collects information about the current logged on user from the domain controller and/or by querying the client machine. Then the IP address of an incoming client request is mapped to a user identity in the domain. If authorization information is also needed, then another realm (LDAP or local) must be created.

Synopsis

Allows you to create and manage Windows SSO realms.

Syntax

`#(config) security windows-sso [subcommands]`

Subcommands

`#(config) security windows-sso create-realm realm_name`
Creates the specified Windows SSO realm.

`#(config) security windows-sso edit-realm realm_name`
Changes the prompt to allow configuration for the specified *realm_name*.

`SGOS#(config windows-sso realm_name) alternate-agent {host hostname | port port_number}`
Specifies the alternate agent hostname and port number.

`SGOS#(config windows-sso realm_name) authorization {realm-name authorization-realm-name | username username | no {authorization-realm-name | username} | self}`
Specifies the realm name, which can be **self**, and username for authorization. **No** clears the realm and username.

`SGOS#(config windows-sso realm_name) cookie {persistent {disable | enable}| verify-ip {disable | enable}}`
Specifies whether to enable persistent or session cookies, and whether to verify the IP address of the cookie.

`SGOS#(config windows-sso realm_name) exit`
Leaves the windows-sso edit-realm mode.

`SGOS#(config windows-sso realm_name) inactivity-timeout seconds`
Specifies the amount of time a session can be inactive before being logged out.

`SGOS#(config windows-sso realm_name) no alternate-agent`
Removes the alternate agent.

`SGOS#(config windows-sso realm_name) primary-agent {host hostname | port port_number}`
Specifies the primary agent hostname and port number.

`SGOS#(config windows-sso realm_name) refresh-time {authorization-refresh seconds | surrogate-refresh seconds}`
Sets the refresh time for authorization and surrogates.

`SGOS#(config windows-sso realm_name) rename new_realm_name`
Renames the current realm to *new_realm_name.*

`SGOS#(config windows-sso realm_name) ssl {enable | disable}`
Enables or disables SSL between the ProxySG and the BCAAA service.

`SGOS#(config windows-sso` *realm_name*`)` **`ssl-device-profile`** *ssl_device_profile_name*
Specifies the device profile to use

`SGOS#(config windows-sso` *realm_name*`)` **`sso-type`** {**`query-client`** | **`query-dc`** | **`query-dc-client`**}
Selects the method of querying: client, domain controller, or both. The default is domain controller.

`SGOS#(config windows-sso` *realm_name*`)` **`test-authentication`** *IP_address*
Tests the Windows SSO and BCAAA configuration to ensure that the `ProxySG` appliance can successfully map an IP address to a user in your Active Directory.

`SGOS#(config windows-sso` *realm_name*`)` **`timeout`** *seconds*
The time allotted for each request attempt. The default is 60 seconds.

`SGOS#(config windows-sso` *realm_name*`)` **`view`**
Displays this realm's configuration.

`SGOS#(config windows-sso` *realm_name*`)` **`virtual-url`** *url*
Specifies the virtual URL to use for this SiteMinder realm. If no URL is specified the global transparent proxy virtual URL is used.

`#(config)` **`security windows-sso delete-realm`** *realm_name*
Deletes the specified Windows SSO realm.

`#(config)` **`security windows-sso view`** [*realm_name*]
Displays the configuration of all Windows SSO realms or just the configuration for *realm_name* if specified.

For More Information

- *SGOS Administration Guide*

Example

```
SGOS#(config) security windows-sso edit-realm test2
SGOS#(config windows-sso test2) ssotype query-client-dc
 ok
SGOS#(config windows-sso test2) exit
```

#(config) security xml

An XML realm uses XML messages to request authentication and authorization information from an HTTP XML service (the XML *responder* that runs on an external server). The XML realm (the XML *requestor*) supports both HTTP GET and HTTP POST methods to request an XML response. The XML messages are based on SOAP 1.2.

The XML responder service accepts XML requests from the ProxySG, communicates with an authentication or authorization server, and responds with the result. When the realm is used to authenticate users, it challenges for Basic credentials. The username and password are then sent to the XML responder to authenticate and authorize the user.

The XML realm can place the username and password in the HTTP headers of the request or in the body of the XML POST request. If the credentials are placed in the HTTP headers, the Web server must do the authentication and the XML service just handles authorization. If credentials are placed in the XML request body, the XML service handles both authentication and authorization.

Synopsis

Allows you to configure and manage XML realms.

Syntax

```
#(config) security xml [subcommands]
```

Subcommands

```
#(config) security xml create-realm realm_name
```
Creates the specified XML realm

```
#(config) security xml delete-realm realm_name
```
Deletes the specified XML realm.

```
#(config) security xml edit-realm realm_name
```
Changes the prompt. See Submodes for details.

```
#(config) security xml view [realm_name]
```
Displays the configuration of all XML realms or just the configuration for *realm_name* if specified.

Submodes

```
#(config) security xml edit-realm realm_name
```

This changes the prompt to:

```
#(config xml realm_name)
```

Commands in the `xml realm_name` mode:

```
#(config xml realm_name) alternate-responder {host | port}
```
Specifies the alternate responder host and port.

```
#(config xml realm_name) alternate-responder path {authenticate
    authenticate_path | authorize authorize_path}
```
Specifies the alternate responder path for authentication and authorization requests.

```
#(config xml realm_name) authorization {default-group-name group-name | username
    use-full-username | realm {none | username | self}}
```
Specifies the default group name, username, and realm for authorization.

```
#(config xml realm_name) connections count
```
Specifies the number of connections to the responder.

`#(config xml realm_name) cookie {persistent {enable | disable} | verify-ip {enable | disable}`
Specifies whether to enable persistent or session cookies, and whether to verify the IP address of the cookie.

`#(config xml realm_name) display-name display_name`
Specifies the display name for this realm.

`#(config xml realm_name) exit`
Exits configure XML-realm mode and returns to configure mode.

`#(config xml realm_name) inactivity-timeout seconds`
Specifies the amount of time a session can be inactive before being logged out.

`#(config xml realm_name) log-out {challenge {enable | disable} | display-time seconds}`
Allows you to challenge the user after log out and define the log out page display time.

`#(config xml realm_name) no alternate-responder`
Removes the alternate-responder.

`#(config xml realm_name) no default-group-name`
Removes the default-group-name.

`#(config xml realm_name) one-time-passwords {enable | disable}`
Allows you to use one-time passwords for authentication. The default is disabled.

`#(config xml realm_name) primary-responder {host | port}`
Specifies the primary responder host and port.

`#(config xml realm_name) primary-responder path {authenticate authenticate_path | authorize authorize_path}`
Specifies the primary responder path for authentication and authorization requests.

`#(config xml realm_name) refresh-time {authorization-refresh seconds | credential-refresh seconds | rejected-credentials-refresh seconds| surrogate-refresh seconds}`
Sets the refresh time for authorization, credential, rejected credentials cache, and surrogates.

`#(config xml realm_name) rename new_realm_name`
Renames this realm to *new_realm_name*.

`#(config xml realm_name) retry count`
Specifies the number of times for the system to retry a request. The default is not to retry a request.

`#(config xml realm_name) server-authentication {none | origin | proxy}`
Enables/disables the forwarding of BASIC credentials of the authenticated user to the origin content server or for proxy authentication. Flush the entries for a realm if the server-authentication value is changed to ensure that the server-authentication value is immediately applied.

You can only choose one server-authentication method:

- If set to **origin**, BASIC credentials are forwarded to an upstream server.
- If set to **proxy**, BASIC credentials are forwarded to an upstream proxy.
- If set to **none**, forwarding BASIC credentials is disabled.

`#(config xml realm_name) timeout seconds`
Specifies the XML request timeout. This is the number of seconds the ProxySG allows for each request attempt before giving up on a server and trying another server. Within a timeout multiple packets can be sent to the server, in case the network is busy and packets are lost. The default request timeout is 10 seconds

`#(config xml realm_name) view`
Displays this realm's configuration.

`#(config xml` *`realm_name`*`)` **`virtual-url`** *`virtual URL`*

Specifies the virtual URL to use for this realm. If no URL is specified the global transparent proxy virtual URL is used.

`#(config xml` *`realm_name`*`)` **`xml`** `{`**`credentials`** `{`**`header`** `|` **`request`**`}` `|` **`request-interested`** `{`**`enable`** `|` **`disable`**`}` `|` **`username`** *`username_parameter`*`}`

Specifies the user credential location and the username parameter. The username parameter is passed in the request when this realm is used for authentication or authorization.

For More Information

- *SGOS Administration Guide*

Example

```
#(config) security xml edit-realm xml14
#(config xml xml14) display-name
 ok
#(config xml xml14) server-authentication origin
 ok
#(config xml xml14) exit
```

#(config) service-groups

Synopsis

Proxy services are defined on the Proxy Services page (**Configuration > Services > Proxy Services**) and are grouped together into predefined service groups based on the type of traffic they handle. Service groups allow you to:

- Intercept and bypass traffic at the service group level
- Create and delete custom service groups

Syntax

```
#(config) service-groups
```

This changes the prompt to:

```
#(config service-groups)
```

Subcommands

`#(config service-groups)` **`bypass-all`** *`service-group`*
Sets all listeners in a service group to bypass.

`#(config service-groups)` **`create`** *`service-group`*
Creates a proxy service group.

`#(config service-groups)` **`delete`** *`service-group`*
Deletes a proxy service group.

`#(config service-groups)` **`exit`**
Returns to the `#(config)` prompt.

`#(config service-groups)` **`intercept-all`** *`service-group`*
Sets all listeners in a service group to intercept.

`#(config service-groups)` **`view`** *`service-group`*
Shows details about a service group. View details about all the service groups by pressing `<enter>`.

For More Information

- *SGOS Administration Guide,* Proxy Services chapter

Example

```
Service Group:  Encrypted
Action:         intercept-all
Services:       HTTPS, IMAPS, POP3S

Service Group:  Interactive
Action:         intercept-all
Services:       Telnet, MS Terminal Services, Shell, SSH, VNC, X Windows

Service Group:  Intranet
Action:         mixed
Services:       Endpoint Mapper, CIFS, Novell GroupWise, Citrix ICA, IMAP,
Kerberos, LDAP, Lotus Notes, LPD, MS SQL Server, MySQL, NFS, Novell NCP, Oracle,
POP3, SMTP, SnapMirror, Sybase SQL
```

#(config) session-monitor

Synopsis

Use this command to configure options to monitor RADIUS accounting messages and to maintain a session table based on the information in these messages.

Syntax

```
#(config) session-monitor
```

This changes the prompt to:

```
#(config session-monitor)
```

Subcommands

`#(config session-monitor)` **`attributes`**
Changes the prompt to allow configuration of session-monitor attributes.

`#(config session-monitor attributes)` **`add`** *`attribute_name`*
Start storing an attribute.

`#(config session-monitor attributes)` **`exit`**
Exit to the session-monitor prompt.

`#(config session-monitor attributes)` **`remove`** *`attribute_name`*
Stop storing an attribute.

`#(config session-monitor attributes)` **`view`**
View the list of attributes being stored.

`#(config session-monitor)` **`cluster disable`**
Disables cluster support.

`#(config session-monitor)` **`cluster enable`**
Enables cluster support. The group address must be set before the cluster can be enabled.

`#(config session-monitor)` **`cluster grace-period`** *`seconds`*
Set the time to keep session transactions in memory while waiting for slave logins. This can be set to allow session table synchronization to occur after the synchronization-delay has expired. The default is 30 seconds; the range is 0 to 2^31-1 seconds.

`#(config session-monitor)` **`cluster no group-address`** *`IP_Address`*
Set or clear (the default) the failover group IP address. This must be an existing failover group address.

`#(config session-monitor)` **`cluster port`** *`port`*
Set the TCP/IP port for the session replication control. The default is 55555.

`#(config session-monitor)` **`cluster synchronization-delay`** *`seconds`*
Set the maximum time to wait for session table synchronization. The default is zero; the range is from 0 to 2 ^31 -1 seconds. During this time evaluation of `$(session.username)` is delayed, so proxy traffic might also be delayed.

`#(config session-monitor)` **`cluster retry-delay`** *`seconds`*
Specify the maximum delay between connection retries. The valid range is 1-1440 minutes.

`#(config session-monitor)` **`disable`**
Disable (the default) session monitoring.

`#(config session-monitor)` **`enable`**
Enable session monitoring.

`#(config session-monitor)` **`max-entries`** *`integer`*
The maximum number of entries in the session table. The default is 500,000; the range is from 1 to 2,000,000. If the table reaches the maximum, additional START messages are ignored.

`#(config session-monitor)` **`radius acct-listen-port`** *`port`*
The port number where the ProxySG listens for accounting messages.

`#(config session-monitor)` **`radius authentication {disable | enable}`**
Enable or disable (the default) the authentication of RADIUS messages using the shared secret. Note that the shared secret must be configured before authentication is enabled.

`#(config session-monitor)` **`radius encrypted-shared-secret`** *`encrypted-secret`*
Specify the shared secret (in encrypted form) used for RADIUS protocol authentication. The secret is decrypted using the configuration-passwords-key.

`#(config session-monitor)` **`radius no shared-secret`**
Clears the shared secret used for RADIUS protocol authentication.

`#(config session-monitor)` **`radius respond {disable | enable}`**
Enable (the default) or disable generation of RADIUS responses.

`#(config session-monitor)` **`radius shared-secret`** *`plaintext_secret`*
Specify the shared secret used for RAIDUS protocol in plaintext.

`#(config session-monitor)` **`timeout`** *`minutes`*
The amount of time before a session table entry assumes a STOP message has been sent. The default is 120 minutes; the range is from 0 to 65535 minutes. Zero indicates no timeout.

`#(config session-monitor)` **`view`**
View the session-monitor configuration.

For More Information

- *SGOS Administration Guide*

Example

```
SGOS#(config) session-monitor
SGOS#(config session-monitor) view
General:
  Status: disabled
  Entry timeout: 120 minutes
  Maximum entries: 500000
  Cluster support: disabled
  Cluster port: 55555
  Cluster group address: none
  Synchronization delay: 0
  Synchronization grace period: 30
Accounting protocol: radius
  Radius accounting:
  Listen ports:
  Accounting: 1813
  Responses: Enabled
  Authentication: Disabled
  Shared secret: ************
```

#(config) sg-client

Synopsis

Replaced by `#(config) proxy-client`. See **`#(config) proxy-client`** on page 260.

#(config) shell

Synopsis

Use this command to configure options for the shell.

Syntax

```
#(config) shell [subcommands]
```

Subcommands

#(config) shell max-connections
Maximum number of shell connections. Allowed values are between 1 and 65535.

#(config) shell no {max-connections | prompt | realm-banner | welcome-banner}
Disables the prompt, realm-banner, welcome-banner, and max connections.

#(config) shell prompt
Sets the prompt that the user sees in the shell. If the string includes white space, enclose the string in quotes.

#(config) shell realm-banner
Sets the realm banner that the user sees when logging into a realm through the shell. If the string includes white space, enclose the string in quotes.

#(config) shell welcome-banner
Sets the welcome banner that the users sees when logging into the shell. If the string includes white space, enclose the string in quotes.

For More Information

- *SGOS Administration Guide*

Example

```
SGOS#(config) shell prompt "Telnet Shell >"
  ok
SGOS#(config) shell welcome-banner "Welcome to the Blue Coat Telnet Shell"
  ok
```

#(config) show

- # **show** on page 78

#(config) smbv2

Synopsis

Configure the CIFS proxy for SMBv2 connections. See **#(config) cifs** on page 147 for configuring settings for SMBv1 connections.

Syntax

`SGOS#(config) smbv2`

This changes the prompt to:

`SGOS#(config smbv2)`

Subcommands

`SGOS#(config smbv2) disable`
Disable protocol-based acceleration for SMBv2 connections. All SMBv2 connections are passed through, allowing the CIFS proxy to accelerate them with byte caching and compression techniques (if enabled for the CIFS service). No object caching is performed on SMBv2 connections.

`SGOS#(config smbv2) downgrade`
Forces the negotiation of SMBv1 for the connection. If this isn't possible (for example, if the client negotiates SMBv2 directly or CIFS SMBv1 protocol acceleration is disabled), the connection is passed through, allowing it to be accelerated with byte caching and compression techniques (if enabled for the CIFS service). No object caching is performed on these connections.

`SGOS#(config smbv2) enable`
Unsigned SMBv2 connections are accelerated with object caching, byte caching (if enabled for the CIFS service), and compression (if enabled). SMBv2 connections that require signing are passed through, allowing the CIFS proxy to accelerate them with byte caching and compression techniques (if enabled).

`SGOS#(config smbv2) exit`
Returns to the (config) submode.

`SGOS#(config smbv2) view {configuration | statistics}`
Views the configuration or statistics for SMBv2.

For More Information

- "Accelerating File Sharing" chapter in the *SGOS Administration Guide*
- **#(config) cifs** on page 147

Example

```
SGOS#(config)smbv2
SGOS#(config smbv2) view configuration
SMBv2:                          Enabled
SGOS#(config smbv2) disable
 ok
SGOS#(config smbv2)view configuration
SMBv2:                          Disabled
SGOS#(config smbv2) exit
SGOS#(config)
```

#(config) smtp

Synopsis

Use this command to configure settings for sending email notification to administrators. Note that this command configures the SMTP server and the sender's email address; the recipient list is configured with the **event-log mail add** command.

Syntax

```
#(config) smtp
```

This changes the prompt to:

```
#(config smtp)
```

Subcommands

`#(config smtp)` **exit**
Exit configure SMTP mode and returns to configure mode.

`#(config smtp)` **from** ***from-address***
Specify the sender's email address; this address displays in the From field for email notifications that the ProxySG sends.

`#(config smtp)` **no server**
Clear the configured SMTP server.

`#(config smtp)` **server** ***domainname | ip-address [port]***
Configure the mail server. You can specify a domain name that resolves to an IPv4 or IPv6 address, or an IPv4 or IPv6 address of the mail server. The default port, if not specified, is 25.

`#(config smtp)` **view**
Show SMTP server and from-address settings.

For More Information

- **#(config) event-log** on page 186

Example

```
SGOS#(config) smtp
SGOS#(config smtp) server mail.test.com
 ok
SGOS#(config smtp) from john.smith@test.com
```

#(config) snmp

Synopsis

Use this command to set SNMP (Simple Network Management Protocol) options for the ProxySG. The ProxySG can be viewed using an SNMP management station and supports MIB-2 (RFC 1213).

Syntax

```
#(config) snmp
```

This changes the prompt to:

```
#(config snmp)
```

Subcommands

`#(config snmp) authentication-failure-traps {enable | disable}`
Enables or disables traps for SNMP protocol authentication failures.

`#(config snmp) create {community` *community_string* `| user` *username*`}`
Creates a new SNMPv1–v2c community or new SNMPv3 user.

`#(config snmp) delete {community` *community_string* `| user` *username*`}`
Deletes an SNMPv1–v2c community string or SNMPv3 user.

`#(config snmp) edit {community` *community_string* `| user` *username*`}`
Allows you to edit an SNMPv1–v2c community's access, traps, and informs, or edit an SNMPv3 user's configuration, access, traps, and informs. See **#(config snmp community <community-string>)** on page 386 and **#(config snmp user <username>)** on page 388.

`# (config snmp) engine-id {default | set` *hexadecimal_string*`}`
Sets the engine ID to the default value or allows you to set it with hexadecimal digits.

`#(config snmp) exit`
Exits configure SNMP mode and returns to configure mode.

`#(config snmp) no {sys-contact | sys-location}`
Clears the system contact string or the system location string.

`#(config snmp) protocol snmpv1 {disable | enable}`
Enables or disables the use of SNMPv1.

`#(config snmp) protocol snmpv2c {disable | enable}`
Enables or disables the use of SNMPv2c.

`#(config snmp) protocol snmpv3 {disable | enable}`
Enables or disables the use of SNMPv3.

`#(config snmp) sys-contact` *string*
Sets the appliance's contact name for display in MIBs.

`#(config snmp) sys-location` *string*
Sets the appliance's location for display in MIBs.

`#(config snmp) test-trap` *string*
Sends a policy test trap with the given text string to test communication. Quotes are required if the message contains whitespace.

`#(config snmp) traps {disable | enable}`
Disables or enables the use of all traps and informs.

`#(config snmp) view`
Displays the SNMP configuration.

`#(config snmp) view {communities | users}`
Displays SNMPv1 and SNMPv2c communities or SNMPv3 users.

For More Information

- *SGOS Administration Guide*
- For details about configuring SNMPv1 and SNMPv2, see **#(config snmp community <community-string>)** on page 386. For details about configuring SNMPv3, see **#(config snmp user <username>)** on page 388.

Example

```
SGOS#(config) snmp
SGOS#(config snmp) authorize-traps
 ok
SGOS#(config snmp) exit
SGOS#(config)
```

#(config snmp community <community-string>)

Synopsis

Use this command to configure community strings for SNMPv1 and SNMPv2c, their access control, and their trap and inform recipients.

Syntax

`#(config snmp) edit community` *community_string*

This changes the prompt to:

`#(config snmp community` *community_string*`)`

Subcommands

`#(config snmp community` *community_string*`) add {inform | trap}`
Adds an SNMPv2c inform receiver or a trap receiver for this community.

`#(config snmp community` *community_string*`) add inform udp` *IP*`[:`*port*`]`
Sends SNMPv2c UDP informs to this IP address.

`#(config snmp community` *community_string*`) add trap {snmpv1 | snmpv2c}`
Adds an SNMPv1 or SNMPv2c trap receiver.

`#(config snmp community` *community_string*`) add trap snmpv1 udp` *IP*`[:`*port*`]`
Sends SNMPv1 UDP traps to this IP address.

`#(config snmp community` *community_string*`) add trap snmpv2c udp` *IP*`[:`*port*`]`
Sends SNMPv2c UDP traps to this IP address.

`#(config snmp community` *community_string*`) authorization access-list`
Enables you to configure a list of allowed source addresses for SNMP requests; changes the prompt to `#(config snmp community access` *community_string*`)`.

`#(config snmp community access` *community_string*`) add {IP | subnet}`
Allows requests from the specified address.

`#(config snmp community access` *community_string*`) clear`
Clears the access list.

`#(config snmp community access` *community_string*`) disable`
Disables the use of the access list and allows requests from all addresses.

`#(config snmp community access` *community_string*`) enable`
Enables use of the access list.

`#(config snmp community access` *community_string*`) exit`
Returns to the `#(config snmp community` *community_string*`)` mode.

`#(config snmp community access` *community_string*`) remove {IP | subnet}`
`Do not allow requests from this address.`

`#(config snmp community access` *community_string*`) view`
Displays the community's access list.

`#(config snmp community` *community_string*`) authorization mode {none | read-only | read-write}`
Allows you to set the read or write access allowed for SNMP requests: none (do not allow any remote access), read-only, or read-write.

`#(config snmp community` *community_string*`) remove {inform | trap}`
Removes an SNMPv2c inform receiver or an SNMPv1 trap receiver.

`#(config snmp community` *`community_string`*`)` **`remove inform udp`** *`IP[:port]`*
Stops sending SNMPv2c UDP informs to this address.

`#(config snmp community` *`community_string`*`)` **`remove trap {snmpv1 | snmpv2c}`**
Removes an SNMPv1 or SNMPv2c trap receiver.

`#(config snmp community` *`community_string`*`)` **`remove trap snmpv1 udp`** *`IP[:port]`*
Stops sending SNMPv1 UDP traps to this address.

`#(config snmp community` *`community_string`*`)` **`remove trap snmpv2c udp`** *`IP[:port]`*
Stops sending SNMPv2c UDP traps to this address.

`#(config snmp community` *`community_string`*`)` **`view`**
Displays the community's authorization, traps, and informs.

For More Information

- *SGOS Administration Guide*
- For general SNMP commands, see **`#(config) snmp`** on page 384. To configure SNMP for SNMPv3, see **`#(config snmp user <username>)`** on page 388.

#(config snmp user <username>)

Synopsis

Use this command to configure users for SNMPv3, their access control, and their trap and inform recipients.

Syntax

`#(config snmp)`**`edit user`** *`username`*

This changes the prompt to:

`#(config snmp user` *`username`*`)`

Subcommands

`#(config snmp user` *`username`*`)` **`add`** {**`inform`** | **`trap`**}
Adds a trap or inform receiver for this user.

`#(config snmp user` *`username`*`)` **`add inform udp`** *`IP`*`[:`*`port`*`]`
Sends SNMPv3 UDP informs to this IP address.

`#(config snmp user` *`username`*`)` **`add trap udp`** *`IP`*`[:`*`port`*`]`
Sends SNMPv3 UDP traps to this IP address.

`#(config snmp user` *`username`*`)` **`authentication`**
Configures the user's authentication settings.

`#(config snmp user` *`username`*`)` **`authentication encrypted_localized_key`** **`<`***`encrypted_key`***`>`**
Enter an encrypted localized key for an engine ID.

`#(config snmp user` *`username`*`)` **`authentication encrypted_passphrase`** `<`*`encrypted_passphrase`*`>`
Enter an encrypted passphrase.

`#(config snmp user` *`username`*`)` **`authentication localized-key`** `<`*`engine_id`*`>` `<`*`key`*`>`
Enter a clear text localized key for an engine ID (in hexadecimal format).

`#(config snmp user` *`username`*`)` **`authentication mode`** {**`md5`** | **`sha`**}
Enable authentication with MD5 or SHA based hashing.

`#(config snmp user` *`username`*`)` **`authentication mode none`**
Disable the use of authentication.

`#(config snmp user` *`username`*`)` **`authentication no localized_key`** `<`*`engine_id`*`>`
Remove a localized key.

`#(config snmp user` *`username`*`)` **`authentication passphrase`** `<`*`passphrase`*`>`
Enter a cleartext passphrase.

`#(config snmp user` *`username`*`)` **`authorization`**
Conigures the access authorized for this user.

`#(config snmp user` *`username`*`)` **`authorization mode`** {**`none`** | **`read-only`** | **`read-write`**}
Allows you to set the read or write access allowed for SNMP requests: none (do not allow any remote access), read-only, or read-write.

`#(config snmp user` *`username`*`)` **`exit`**
Returns to `(config snmp)` mode.

`#(config snmp user` *`username`*`)` **`privacy`**
Configures the user's privacy settings.

`#(config snmp user` *`username`*`)` **`privacy encrypted_localized_key`** `<`*`engine_id`*`>` `<`*`encrypted_key`*`>`
Enter an encrypted localized key for the engine ID.

`#(config snmp user` *`username`*`)` **`privacy encrypted_passphrase`** `<`*`encrypted_passphrase`*`>`
Enter an encrypted passphrase.

`#(config snmp user` *`username`*`)` **`privacy localized_key`** `<`*`engine_id`*`>` `<`*`key`*`>`
Enter a clear text localized key for an engine ID (in hexadecimal format).

`#(config snmp user` *`username`*`)` **`privacy mode`** `{`**`none | aes | des`**`}`
Set the encryption mode to none (disable the use of privacy), or enable privacy with AES or DES based encryption.

`#(config snmp user` *`username`*`)` **`privacy no localized_key`** `<`*`engine_id`*`>`
Remove a localized key.

`#(config snmp user` *`username`*`)` **`privacy passphrase`** `<`*`passphrase`*`>`
Enter a cleartext passphrase.

`#(config snmp user` *`username`*`)` **`remove inform udp`** *`IP[:port]`*
Stop sending SNMPv3 UDP informs to this IP address.

`#(config snmp user` *`username`*`)` **`remove trap udp`** *`IP[:port]`*
Stop sending SNMPv3 UDP traps to this IP address.

`#(config snmp user username)` **`view`**
Displays the user's configuration, authorization, traps, and informs.

For More Information

- *SGOS Administration Guide*
- For general SNMP commands, see **`#(config) snmp`** on page 384. To configure SNMP for SNMPv1 and SNMPv2c, see **`#(config snmp community <community-string>)`** on page 386.

#(config) socks-gateways

Synopsis

Use this command to set the SOCKS gateways settings.

Syntax

```
#(config) socks-gateways
```

This changes the prompt to:

```
#(config socks-gateways)
```

Subcommands

#(config socks-gateways) create *gateway_alias gateway_host SOCKS_port* [**group**=*group-alias*] [**version=**{**4** | **5** [**user=***username* {**password**=*password* | **encrypted-password**=*encrypted-password}*]]

Creates a SOCKS gateway.

Note: The SOCKS compression feature is deprecated, as a more advanced version of this functionality is now available as part of the Application Delivery Network features. Refer to the Configuring an Applicant Delivery Network chapter in the *Blue Coat SGOS 6.2 Administration Guide* for instructions on how to configure and use these features.

#(config socks-gateways) create {**gateway** | **group** *group_name*}

#(config socks-gateways) delete {**all** | **gateway** *gateway_alias* | **group** *group_name*}

Deletes a SOCKS gateway or group.

#(config socks-gateways) destroy-old-passwords

Destroys any cleartext passwords left after an upgrade.

#(config socks-gateways) edit *gateway_alias*

Changes the prompt. See **#(config socks-gateways gateway_alias)** on page 392.

#(config socks-gateways) edit *group_alias*

Changes the prompt. See**#(config socks-gateways group_alias)** on page 394.

#(config socks-gateways) exit

Exits configure socks-gateways mode and returns to configure mode.

#(config socks-gateways) failure-mode {**open** | **closed**}

Sets the default failure mode (that can be overridden by policy).

#(config socks-gateways) host-affinity http {**default** | **none** | **client-ip-address** | **accelerator-cookie**} *gateway_or_group_alias*

Selects a host affinity method for HTTP. If a gateway or group alias is not specified for the `accelerator-cookie`, `client-ip-address`, or `none` options, the global default is used. Use the `default` option to specify default configurations for all the settings for a specified gateway or group.

`#(config socks-gateways)` **`host-affinity ssl`** {**`default`** | **`none`** | **`client-ip-address`** | **`accelerator-cookie`** | **`ssl-session-id`**} *`gateway_or_group_alias`*
Selects a host affinity method for SSL. If a gateway or group alias is not specified for the `accelerator-cookie`, `client-ip-address`, `none`, or `ssl-session-id` options, the global default is used. Use the `default` option to specify default configurations for all the settings for a specified gateway or group.

`#(config socks-gateways)` **`host-affinity other`** {**`default`** | **`client-ip-address`** | **`none`**} *`gateway_or_group_alias`*
Selects a host affinity method (non-HTTP or non-SSL). If a gateway or group alias is not specified for the `client-ip-address`, or `none` options, the global default is used. Use the `default` option to specify default configurations for all the settings for a specified gateway or group.

`#(config socks-gateways)` **`host-affinity timeout`** *`minutes`*
Set the timeout for host affinity in minutes.

`#(config socks-gateways)` **`load-balance gateway`** {**`default`** | **`none`** | **`round-robin`** | **`least-connections`**} *`gateway_alias`*
Selects a host affinity method (non-HTTP or non-SSL). If a gateway alias is not specified for the `client-ip-address`, or `none` options, the global default is used. Use the `default` option to specify default configurations for all the settings for a specified gateway.

`#(config socks-gateways)` **`load-balance group`** {**`default`** | **`none`** | **`domain-hash`** | **`url-hash`** | **`round-robin`** | **`least-connections`**} *`group_alias`*

`#(config socks-gateways)` **`no path`**
Clears network path to download SOCKS gateway settings.

`#(config socks-gateways)` **`path`** *`url`*
Specifies the network path to download SOCKS gateway settings.

`#(config socks-gateways)` **`sequence`** {**`add`** | **`demote`** | **`promote`** | **`remove`**} *`gateway_alias`*
Adds an alias to the end of the default failover sequence.

`socks-gateways)` **`sequence clear`**
Clears the default failover sequence.

`#(config socks-gateways)` **`view`**
Displays all SOCKS gateways.

For More Information

- *SGOS Administration Guide*

Example

```
SGOS#(config) socks-gateways
SGOS#(config socks-gateways) failure-mode open
 ok
SGOS#(config socks-gateways) exit
SGOS#(config)
```

#(config socks-gateways *gateway_alias)*

Synopsis

These commands allow you to edit the settings of a specific SOCKS gateway.

Syntax

```
#(config) socks-gateways
```

This changes the prompt to:

```
#(config socks-gateways)
edit gateway_alias
```

This changes the prompt to:

```
#(config socks-gateways gateway_alias)
```

Subcommands

`#(config socks-gateways gateway_alias)` **`encrypted-password`**
Changes the version 5 encrypted password.

`#(config socks-gateways gateway_alias)` **`exit`**
Exits configure socks-gateways *`gateway_alias`* mode and returns to configure socks-gateways mode.

`#(config socks-gateways gateway_alias)` **`host`**
Changes the host name.

`#(config socks-gateways gateway_alias)` **`host-affinity http {accelerator-cookie | client-ip-address | default | none}`**
Changes the host affinity method (HTTP) for this host.

`#(config socks-gateways gateway_alias)` **`host-affinity other {client-ip-address | default | none}`**
Changes the host affinity other method for this host.

`#(config socks-gateways gateway_alias)` **`host-affinity ssl {accelerator-cookie | client-ip-address | default | ssl-session-id | none}`**
Changes the host affinity method (SSL) for this host.

`#(config socks-gateways gateway_alias)` **`load-balance {default | least-connections | round-robin | none}`**
Changes the load balancing method.

`#(config socks-gateways gateway_alias)` **`no {password | username}`**
Optional, and only if you use version 5. Deletes the version 5 password or username.

`#(config socks-gateways gateway_alias)` **`password`**
Optional, and only if you use version 5. Changes the version 5 password. If you specify a password, you must also specify a username.

`#(config socks-gateways gateway_alias)` **`port`**
Changes the SOCKS port.

`#(config socks-gateways gateway_alias)` **`request-compression`**
Changes the SOCKS port to request compression.

`#(config socks-gateways gateway_alias)` **`user`**
Optional, and only if you use version 5. Changes the version 5 username. If you specify a username, you must also specify a password.

`#(config socks-gateways` *`gateway_alias`*`)` **`version {4 | 5}`**
Changes the SOCKS version.

`#(config socks-gateways` *`gateway_alias`*`)` **`view`**
Shows the current settings for this SOCKS gateway.

For More Information

- *SGOS Administration Guide*

Example

```
SGOS#(config) socks-gateways
SGOS#(config socks-gateways) edit testgateway
SGOS#(config socks-gateways testgateway) version 5
 ok
SGOS#(config socks-gateways testgateway) exit
SGOS#(config socks-gateways) exit
SGOS#(config)
```

#(config socks-gateways *group_alias)*

Synopsis

These commands allow you to edit the settings of a specific SOCKS gateway group.

Syntax

```
#(config) socks-gateways
```

This changes the prompt to:

```
#(config socks-gateways) create host_alias hostname protocol=port
group=group_alias
```

```
#(config socks-gateways) edit group_alias
```

This changes the prompt to:

```
#(config socks-gateways group_alias)
```

Subcommands

`#(config socks-gateways group_alias) add`
Adds a new group.

`#(config socks-gateways group_alias) exit`
Exits `#(config socks-gateways group_alias)` mode and returns to `#(config socks-gateways)` mode.

`#(config socks-gateways group_alias) host-affinity http {accelerator-cookie | client-ip-address | default | none}`
Changes the host affinity method (HTTP) for this group.

`#(config socks-gateways group_alias) host-affinity other {client-ip-address | default | none}`
Changes the host affinity other method for this host.

`#(config socks-gateways group_alias) host-affinity ssl {accelerator-cookie | client-ip-address | default | ssl-session-id | none}`
Changes the host affinity method (SSL) for this group.

`#(config socks-gateways group_alias) load-balance method {default | domain-hash | least-connections | none | round-robin | url-hash}`
Changes the load balancing method.

`#(config socks-gateways group_alias) remove`
Removes an existing group.

`#(config socks-gateways group_alias) view`
Shows the current settings for this SOCKS gateway.

For More Information

- *SGOS Administration Guide*

Example

```
SGOS#(config) socks-gateways
SGOS#(config socks-gateways) edit test_group
SGOS#(config socks-gateways test_group) load-balance hash domain
 ok
SGOS#(config socks-gateways test_group) exit
SGOS#(config socks-gateways) exit
SGOS#(config)
```

#(config) socks-machine-id

Synopsis

Use this command to set the machine ID for SOCKS.

If you are using a SOCKS server for the primary or alternate gateway, you must specify the ProxySG machine ID for the Identification (Ident) protocol used by the SOCKS gateway.

Syntax

`#(config)` **`socks-machine-id`** *`machine_id`*

Indicates the machine ID for the SOCKS server.

Example

```
SGOS#(config) socks-machine-id 10.25.36.47
 ok
```

#(config) socks-proxy

Synopsis

Use this command to configure a SOCKS proxy on anProxySG. Only one server is permitted per ProxySG. Both SOCKSv4 and SOCKSv5 are supported by Blue Coat, and both are enabled by default.

Note that the version of SOCKS used is only configurable through policy. For example, to use only SOCKSv5:

```
<proxy>
  socks.version=4 deny
```

Syntax

`#(config)` **`socks-proxy`**

Subcommands

`#(config)` **`socks-proxy accept-timeout`** *`seconds`*
Sets maximum time to wait on an inbound BIND.

`#(config)` **`socks-proxy connect-timeout`** *`seconds`*
Sets maximum time to wait on an outbound CONNECT.

`#(config)` **`socks-proxy max-connections`** *`num_connections`*
Sets maximum allowed SOCKS client connections.

`#(config)` **`socks-proxy max-idle-timeout`** *`seconds`*
Specifies the minimum timeout after which SOCKS can consider the connection for termination when the max connections are reached.

`#(config)` **`socks-proxy min-idle-timeout`** *`seconds`*
Specifies the max idle timeout value after which SOCKS should terminate the connection.

`#(config)` **`socks-proxy pa-customer-id`** ***`customer_id`***
Validates the license for the specified customer. (The customer_id is the Customer ID number you took from the **About** t ab on the PA client. Use **`socks-proxy pa-customer-id 0`** to disable the license.

For More Information

- *SGOS Administration Guide*

Example

```
SGOS#(config) socks-proxy accept-timeout 120
 ok
```

#(config) ssh-console

Synopsis

Configures the SSH host and client keys. This CLI command also sets global options, such as the welcome banner for all SSH Consoles on the system.

To create and edit additional SSH console services, see **#(config ssh-console)** on page 243.

Syntax

```
#(config) ssh-console
```

This changes the prompt to:

```
#(config ssh-console)
```

Subcommands

`#(config ssh-console)` **`create host-keypair {sshv1 | sshv2 | <Enter>}`**
Creates a host-keypair for the SSH console of the specified version.

`#(config ssh-console)` **`delete client-key`** *`username key_id`*
Deletes the client key with the specified username and key ID.

`#(config ssh-console)` **`delete legacy-client-key`** *`key_id`*
Deletes the legacy client key.

`#(config ssh-console)` **`delete director-client-key`** *`key_id`*
Deletes the Director client key.

`#(config ssh-console)` **`delete host-keypair {sshv1 | sshv2 | <Enter>}`**
Deletes the specified host keypair.

`#(config ssh-console)` **`inline {client-key <`***`eof`***`> | director-client-key <`***`eof`***`> | sshv2-welcome-banner <`***`eof`***`>}`**
Allows you use the inline commands to add a client key, a Director client key, or a banner for those logging to the ProxySG using SSHv2.

`#(config ssh-console)` **`no sshv2-welcome-banner`**
Disables the welcome banner.

`#(config ssh-console)` **`exit`**
Returns to the `#(config)` prompt.

`#(config ssh-console)` **`view {client-key | director-client-key | host-public-key | user-list | versions-enabled}`**
Views the SSH console parameters.

For More Information

- *SGOS Administration Guide*
- **#(config ssh-console)** on page 243

Example

`#(config ssh-console) view versions-enabled`

SSHv2 is enabled.

#(config) ssl

Synopsis

Use this command to configure HTTPS termination, including managing certificates, both self-signed and those from a Certificate Signing Authority (CSA).

To configure HTTPS termination, you must complete the following tasks:

- Configure a keyring
- Configure the SSL client
- Configure the HTTPS service

Note: To do these steps, you must have a serial or SSH connection; you cannot use Telnet.

Syntax

```
#(config) ssl
```

This changes the prompt to:

```
#(config ssl)
```

Subcommands

`#(config ssl) ccr-renegotiation-list {clear | max-entries <number> | view}`
Manages the Client Certificate Requested list.

`clear`: Removes all entries in the list.

`max-entries <number>`: Specifies the maximum number of entries allowed in the list. The maximum value is 10000, the minimum is 0, and the default is 1000.

`view`: Displays all the entries in the list.

`#(config ssl) create ccl list_name`
Creates a list to contain CA certificates.

`#(config ssl) create certificate keyring_id [attribute_value] [attribute_value]`
Creates a certificate. Certificates can be associated with a keyring.

You can create a self-signed certificate two ways: interactively or non-interactively.

Director uses non-interactive commands in profiles and overlays to create certificates.

`#(config ssl) create crl crl_id`
Create a Certificate Revocation List.

`#(config ssl) create fips {ccl list_name | keyring {no-show <keyring_id> [key_length] | show <keyring_id> [key_length] | show-director <keyring_id> [key_length]} | ssl-device-profile <device_profile_name> [keyring]}`
Create FIPS compliant PKI elements.

`#(config ssl) create keylist list_name`
Create a keylist with no keyrings with the specified name.

`#(config ssl) create keyring {show | show-director | no-show} keyring_id [key_length]`
Creates a keyring, with a keypair, where:

`show`: Keyrings created with this attribute are displayed in the `show configuration` output, meaning that the keyring can be included as part of a profile or overlay pushed by Director.

`show-director`: Keyrings created with this attribute are part of the `show configuration` output if the CLI connection is secure (SSH/RSA) and the command is issued from Director.

`no-show`: Keyrings created with this attribute are not displayed in the show configuration output and cannot be part of a profile. The `no-show` option is provided as additional security for environments where the keys will never be used outside of the particular ProxySG.

`#(config ssl)` **`create ssl-device-profile`** *<SSL_device_profile_name>* [*keyring*]
Creates an SSL device profile of the specified name and keyring.

`#(config ssl)` **`create signing-request`** *keyring_id*
Creates a certificate signing request. The request must be associated with a keyring.

You can create a signing request two ways: interactively or non-interactively.

Director uses non-interactive commands in profiles and overlays to create signing requests.

`#(config ssl)` **`create ssl-client`** *ssl_client_name*
Associates the SSL client with a keyring. Only the default is permitted.

`#(config ssl)` **`delete ca-certificate`** *name*
Deletes a CA-certificate from the ProxySG.

`#(config ssl)` **`delete ccl`** *list_name*
Deletes a CCL list from the ProxySG.

`#(config ssl)` **`delete certificate`** *keyring_id*
Deletes the certificate associated with a keyring.

`#(config ssl)` **`delete crl`** *list_name*
Deletes the specified Certificate Revocation List.

`#(config ssl)` **`delete external-certificate`** *name*
Deletes an external certificate from the ProxySG.

`#(config ssl)` **`delete keylist`** *list_name*

Deletes the specified keylist. Keyrings associated with the keylist are not deleted. If the keylist is used in policy, the keylist cannot be deleted and generates an error.

`#(config ssl)` **`delete keyring`** *keyring_id*
Deletes a keyring, with a keypair.

`#(config ssl)` **`delete keyring [force]`** *keyring_id*
Deletes a keyring. The **`force`** option removes the keyring from all keylists using the keyring, and then deletes the keyring.

`#(config ssl)` **`delete signing-request`** *keyring_id*
Deletes a certificate signing request.

`#(config ssl)` **`delete ssl-client`** *ssl_client_name*
Deletes an SSL client.

`#(config ssl)` **`delete ssl-device-profile`** *ssl_device_profile_name*
Deletes an SSL device profile.

`#(config ssl)` **`edit ccl`** *list_name*
Changes the prompt. See **`#(config ssl ccl list_name)`** on page 405.

`#(config ssl)` **`edit crl`** *crl_id*
Changes the prompt. See **`#(config ssl crl crl_list_name)`** on page 406.

`#(config ssl)` **`edit keylist`** *list_name*
Enables you to configure the keylist parameters; changes the prompt to `#(config ssl keylist` *list_name*).

`#(config ssl keylist` *list_name*`)` **`add`** *keyring_id*
Adds the specified keyring to the keylist.

`#(config ssl keylist` *list_name)* **`remove`** *keyring_id*
Removes the specified keyring from the keylist.

`#(config ssl keylist` *list_name*`)` **`clear`**
Removes all keyrings from the keylist.

`#(config ssl keylist` *list_name*`)` **`extractor`** *extractor_string*
Set the extractor pattern for the keyring. The extractor supports substitutions from all attributes of Subject, Issuer, SubjectAltName, IssuerAltName, and SerialNumber certificate fields. The default extractor value is $(subject.CN); many other subject attributes are recognized, among them OU, O, L, ST, C, and DC. Field indexes can be used in substitutions on a group name or attribute; for example `$(SubjectAltName.DNS.1)`.

`#(config ssl keylist` *list_name*`)` **`view [verbose]`**
Displays the keylist extractor as well as the keyring IDs and their respective extractor values. Use **`verbose`** to display the certificate field values of the keylist.

`#(config ssl keylist` *list_name*`)` **`exit`**
Returns to the `#(config ssl)` mode.

`#(config ssl)` **`edit ssl-device-profile`** *profile_name*
Changes the prompt. See **`#(config ssl-device-profile profile_name)`** on page 407

`#(config ssl)` **`edit ssl-client`** *ssl_client_name*
Changes the prompt. Only default is permitted. See **`#(config ssl client ssl_client_name)`** on page 409.

`#(config ssl)` **`exit`**
Exits configure ssl mode and returns to configure mode.

`#(config ssl)` **`force-secure-renegotiation {enable | disable}`**
Enabling this makes the Proxy perform strict secure renegotiation only for all SSL connections.

Note: By default this feature is disabled. If this feature is enabled, all secure communications with Blue Coat servers will fail. If you enable this setting, the ProxySG appliance performs strict secure renegotiation only for all SSL connections. Currently, Firefox 3.6.8 supports the new secure renegotiation. Internet Explorer 8.0 does not support this feature and will fail if the customer policy is set to do SSL-Intercept.

`#(config ssl)` **`inline ca-certificate`** *name eof*
Imports a CA certificate.

`#(config ssl)` **`inline certificate`** *keyring_id eof*
Imports a certificate.

Note: Using the `inline certificate` command, you can associate a certificate chain with a keyring. You must paste all associated intermediate certificates after the server certificate. The maximum character count for importing a certificate chain and associating it with a keyring is 7999.

`#(config ssl)` **`inline crl`** *list_name eof*
Imports a Certificate Revocation List.

`#(config ssl)` **`inline external-certificate`** *name eof*
Imports a certificate without the corresponding private key.

`#(config ssl)` **`inline fips ca-certificate`** *<name>*`[`*eof_marker*`]`
Install a FIPS compliant Certificate Authority certificate.

`#(config ssl)` **`inline fips external-certificate`** *<name>*`[`*eof_marker*`]`
Install a FIPS compliant certificate without a corresponding private key.

`#(config ssl) inline fips keyring {no-show`
`<keyring_id>[<password>|<""><eof_marker> | show`
`<keyring_id>[<password>|<""><eof_marker> | show-director`
`<keyring_id>[<password>|<""><eof_marker>}`
Install a FIPS compliant keyring with unshowable, showable, or director showable key pairs.

`#(config ssl) inline keylist <list_name> <extractor_string>`
Imports a keylist. Each keyring ID must be listed on independent lines. If a keylist with the same name already exists, it will be replaced with the new information.

`#(config ssl) inline keyring {show | show-director | no-show} keyring_id [password] eof`
Imports a keyring, where:

`show`: Private keys associated with keyrings created with this attribute can be displayed in the CLI or included as part of a profile or overlay pushed by Director.

`show-director`: Keyrings created with this attribute are part of the `show configuration` output if the CLI connection is secure (SSH/RSA) and the command is issued from Director.

`no-show`: Keyrings created with this attribute are not displayed in the `show configuration` output and cannot be part of a profile. The `no-show` option is provided as additional security for environments where the keys will never be used outside of the particular ProxySG.

`password`: The password for the keyring.

`eof`: End-of-file marker. This can be anything, as long as it doesn't also appear in the inline text. (If the *`eof`* appears in the inline text, the `inline` command completes at that point.)

Note: The following keyrings cannot be added to keylists: `default`, `passive-attack-protection`, `config-passwords`, and `default-untrusted`.

`#(config ssl) inline signing-request keyring_id eof`
Imports the specified signing request.

`#(config ssl) intermediate-cert-cache`
Changes the prompt. See **`#(config ssl icc)`** on page 411

`#(config ssl) keydata-path <url>`
Sets the path for keyrings and keylists to import.

`#(config ssl) ocsp`
Changes the prompt. See **`#(config ssl ocsp)`** on page 413

`#(config ssl) proxy client-cert-ccl {ccl_list_name | all | none}`
Specifies the CCL to be used for the client. The default is `all`.

`#(config ssl) proxy issuer-keyring keyring_name`
Specifies the keyring to be used for SSL interception.

`#(config ssl) proxy preserve-untrusted {enable | disable}`
When this feature is enabled, if an OCS presents a certificate to the ProxySG that is not signed by a trusted Certificate Authority (CA), the ProxySG presents the browser with a certificate that is signed by its untrusted issuer keyring. A warning message is displayed to the user, and they can decide to ignore the warning and visit the Website or cancel the request. The default value is disable.

`#(config ssl) proxy server-cert-ccl {ccl_list_name | all}`
Specifies the CCL to be used for the server. The default is `browser-trusted`.

`#(config ssl) proxy untrusted-issuer-keyring <untrusted-issuer-keyring>`
Specifies the keyring used for signing emulated server certificates when preserving an untrusted OCS certificate. The default value is `default-untrusted`.

`#(config ssl) request-appliance-certificate`
Generates an appliance certificate.

`#(config ssl)` **`ssl-nego-timeout`** *`seconds`*
Configures the SSL-negotiation timeout period. The default is 300 seconds.

`#(config ssl)` **`view appliance-certificate-request`**
Displays the appliance certificate request generated by the `request-appliance-certificate` command.

`#(config ssl)` **`view ca-certificate`** *`name`*
Displays the Certificate Authority certificate.

`#(config ssl)` **`view ccl`**
Displays the CA-certificate lists.

`#(config ssl)` **`view certificate`** *`keyring_id`*
Displays the certificate.

`#(config ssl)` **`view crl`** [*`list_name`*]
Displays the specified Certificate Revocation List.

`#(config ssl)` **`view external-certificate`** *`name`*
Displays the external certificate.

`#(config ssl)` **`view keypair`** {**`des`** | **`des3`** | **`unencrypted`**} *`keyring_id`* | *`keyring_id`*}
Displays the keypair. If you want to view the keypair in an encrypted format, you can optionally specify `des` or `des3` before the keyringID. If you specify either `des` or `des3`, you are prompted for the challenge entered when the keyring was created.

`#(config ssl)` **`view keyring`** [*`keyring_id`* | **`unreferenced`** | **`expiring-in`** <*`n`*>]
Displays the keyring, where:

`keyring_id`: Displays the certificate subject, serial number, issuer, and all keylists that the specified keyring is a member of.

`unreferenced`: Lists all the keyrings that are not referenced anywhere else in the configuration or in policy.

`expiring-in` <*`n`*>: Lists all keyrings with certificates expiring in a specified <n> days. To display all keyrings with expired certificates, use the following command:
`#(config ssl)` **`view keyring expiring-in`** *`0`*

`#(config ssl)` **`view keyring`** [*`keyring_id`*]
Displays the keyring.

`#(config ssl)` **`view ocsp`**
Displays SSL OCSP configuration.

`#(config ssl)` **`view proxy`**
Displays SSL proxy configuration.

`#(config ssl)` **`view signing-request`** *`keyring_id`*
Displays the certificate signing request.

`#(config ssl)` **`view ssl-client`**
Displays summary information of SSL clients.

`#(config ssl)` **`view ssl-device-profile`**
Displays SSL device profile.

`#(config ssl)` **`view ssl-nego-timeout`**
Displays SSL negotiation timeout period status summary.

`#(config ssl)` **`view summary`** {**`ca-certificate`** | **`external-certificate`**} [*`name`*]
Displays a summary for all CA-certificate or external-certificate commands, or for the certificate name specified.

Related Commands

`#(config) load crl crl_list`
Loads the specified CRL list.

`#(config) load keydata [<passphrase>]`
Loads the keyrings and keylists from the location specified with **keydata-path**.

For More Information

- *SGOS Administration Guide*

Examples

```
SGOS#(config) ssl
SGOS#(config ssl) create keyring show keyring id [key length]
 ok
SGOS#(config ssl) view keyring keyring id
KeyringID: default
Is private key showable? yes
Have CSR? no
Have certificate? yes
Is certificate valid? yes
CA: Blue Coat SG810
Expiration Date: Jan 23 23:57:21 2013 GMT
Fingerprint: EB:BD:F8:2C:00:25:84:02:CB:82:3A:94:1E:7F:0D:E3
SGOS#(config ssl) exit
SGOS#(config)
```

Create a self-signed SSL certificate:

```
SGOS#(config) ssl
SGOS#(config ssl) create certificate keyring-id cn bluecoat challenge test c US
state CA company bluecoat
```

#(config ssl ccl *list_name)*

Synopsis

Allows you to edit the CCL parameters.

Syntax

```
#(config) ssl
```

This changes the prompt to:

```
#(config ssl) edit ccl list_name
```

This changes the prompt to:

```
#(config ssl ccl list_name)
```

Subcommands

`#(config ssl ccl list_name) add ca_certificate_name`
Adds a CA certificate to this list. (The CA certificate must first be imported in configure ssl mode.)

`#(config ssl ccl list_name) exit`
Exits *`configure ssl ccl list_name`* mode and returns to ssl configure mode.

`#(config ssl ccl list_name) remove ca_certificate_name`
Removes a CA certificate from the specified list.

`#(config ssl ccl list_name) view`
Shows a summary of CA certificates in this list.

For More Information

- *SGOS Administration Guide*

Example

```
SGOS#(config) ssl
SGOS#(config ssl) edit ccl list_name
SGOS#(config ssl ccl list_name) add CACert1
 ok
SGOS#(config ssl ccl list_name) exit
SGOS#(config ssl) exit
SGOS#(config)
```

#(config ssl crl *crl_list_name)*

Synopsis

Allows you to edit the specified Certificate Revocation List name.

Syntax

```
#(config) ssl
```

This changes the prompt to:

```
#(config ssl)
```

```
edit crl crl_list_name
```

This changes the prompt to:

```
#(config ssl crl crl_list_name)
```

Subcommands

`#(config ssl crl crl_list_name)` **`exit`**
Exits `configure ssl crl crl_list_name` mode and returns to ssl configure mode.

`#(config ssl crl crl_list_name)` **`inline`** *`eof_marker`*
Imports a Certificate Revocation List.

`#(config ssl crl crl_list_name)` **`load`** *`crl`*
Downloads the specified Certificate Revocation List.

`#(config ssl crl crl_list_name)` **`path`** *`crl`*
Specifies the network path to download the specified Certificate Revocation List.

`#(config ssl crl crl_list_name)` **`view`**
View the specified Certificate Revocation List.

For More Information

- *SGOS Administration Guide*

#(config ssl-device-profile *profile_name*)

Synopsis

Allows you to create or edit an SSL device profile.

Syntax

```
#(config) ssl
```

This changes the prompt to:

```
#(config ssl)
```

```
edit ssl-device-profile profile_name
```

This changes the prompt to:

```
#(config device-profile profile_name)
```

Subcommands

`#(config device-profile` *`profile_name`*`)` **`cipher-suite`** *`cipher-suite`*
Configures device authentication profile cipher suites. If you press <enter>, you can see the list of available ciphers. The default is AES256-SHA. You can choose more than one cipher suite.

`#(config device-profile` *`profile_name`*`)` **`ccl`** *`ccl_name`*
Configures the device authentication profile CCL.

`#(config device-profile` *`profile_name`*`)` **`device-id`** *`device_ID`*
Configure device authentication profile of the specific device ID.

`#(config device-profile` *`profile_name`*`)` **`exit`**
Returns to the `# (config ssl)` prompt.

`#(config device-profile` *`profile_name`*`)` **`keyring-id`** *`keyring_ID`*
Configures the device authentication profile in the specified keyring.

`#(config device-profile` *`profile_name`*`)` **`no keyring-id`** *`keyring_ID`*
Clears the SSL device profile keyring ID.

`#(config device-profile` *`profile_name`*`)` **`protocol`** {**`sslv2`** | **`sslv3`** | **`tlsv1`** | **`tlsv1.1`** | **`tlsv1.2`**}
Specifies the protocol to use.

`#(config device-profile` *`profile_name`*`)` **`verify-peer`** {**`enable`** | **`disable`**}
Enables or disables device authentication peer verification.

`#(config device-profile` *`profile_name`*`)` **`view`**

For More Information

- *SGOS Administration Guide*

Example

```
#(config device-profile test1) view
Name: test1
Usable for: client
Keyring:
CCL: browser-trusted
Device-id: $(subject.CN)
Cipher suite: rc4-sha
Protocol: TLSv1TLSv1.1TLSv1.2
Verify-peer: enabled
```

#(config ssl client *ssl_client_name)*

Synopsis

Allows you to edit the SSL client parameters. Only the default is permitted.

Syntax

```
#(config) ssl
```

This changes the prompt to:

```
#(config ssl)
edit ssl-client ssl_default_client_name
```

This changes the prompt to:

```
#(config ssl ssl_default_client_name)
```

Subcommands

`#(config ssl ssl_default_client_name) ccl {ccl_name | all}`
Configures the CA Certificate List to use.

#(config ssl *ssl_default_client_name*) **`cipher-suite`**
Specifies the cipher suite to use. The default is to use all cipher suites. If you want to change the default, you have two choices:

- interactive mode
- non-interactive mode

Director uses non-interactive commands in profiles and overlays to create cipher suites.

The optional *`cipher-suite`* refers to the cipher-suites you want to use, space separated, such as `rc4-md5 exp-des-cbc-sha`. If you want to use the interactive mode, do not specify a cipher suite.

`#(config ssl ssl_default_client_name) exit`
Exits `configure ssl ssl-client ssl_default_client_name` mode and returns to `ssl configure` mode.

`#(config ssl ssl_default_client_name) keyring-id keyring_id`
Configures SSL client keyring id.

`#(config ssl ssl_default_client_name) no keyring-id`
Clears the keyring-id.

`#(config ssl ssl_default_client_name) protocol {sslv2 | sslv3 | tlsv1 | tlsv1.1 | tlsv1.2}}`
Configures SSL client protocol version.

`#(config ssl ssl_default_client_name) view`
Displays the SSL client details.

For More Information

- *SGOS Administration Guide*

Example

```
SGOS#(config) ssl
SGOS#(config ssl) edit ssl-client ssl_default_client_name
SGOS#(config ssl ssl-client ssl_default_client_name) cipher-suite rc4-md5
exp-des-cbc-sha
 ok
SGOS#(config ssl ssl-client ssl_default_client_name) exit
SGOS#(config ssl) exit
SGOS#(config)
```

#(config ssl icc)

Synopsis

Allows you to configure and view intermediate certificate cache (ICC) settings and statistics on the ProxySG appliance.

Syntax

```
#(config) ssl
```

This changes the prompt to:

```
#(config ssl)
#(config ssl) intermediate-cert-cache
```

This changes the prompt to:

```
#(config ssl icc)
```

Subcommands

`#(config ssl icc) clear-cache`
Clears the intermediate CA certificates that are currently stored on the appliance.

`#(config ssl icc) enable`
Enables the caching of intermediate CA certificates on the ProxySG appliance.

`#(config ssl icc) exit`
Exits the `config ssl icc` prompt and returns to the `config ssl` prompt.

`#(config ssl icc) disable`
Simultaneously disables the caching of intermediate CA certificates and clears the existing cache on the ProxySG appliance.

`#(config ssl icc) view status`
Displays the current status of the intermediate certificate cache, including usage statistics and the number of stored intermediate CA certificates.

`#(config ssl icc) view certificate {detail` *certificate_name* `| summary | summary` *certificate_name*`}`
You can view various details about the certificates that have been cached on the appliance.

For More Information

- *SGOS Administration Guide*

Example

```
SGOS#(config) ssl
SGOS#(config ssl) intermediate-cert-cache
SGOS#(config ssl icc) enable
 ok
SGOS#(config ssl icc) view status
Intermediate Certificate Cache
    Caching: enabled
    Number of cached certificates: 4
    Number of new certificates:    2
```

```
    Number of cache hits:          14
SGOS#(config ssl icc) clear-cache
 ok
SGOS#(config ssl icc) exit
```

#(config ssl ocsp)

Synopsis

Allows you to configure Online Certificate Status Protocol (OCSP) settings.

Syntax

```
#(config) ssl
```

This changes the prompt to:

```
#(config ssl)
#(config ssl)ocsp
```

This changes the prompt to:

```
#(config ssl ocsp)
```

Subcommands

`#(config ssl ocsp)`**`create`** *`responder_name`*
Creates a responder.

`#(config ssl ocsp)`**`default`** *`responder_name`*
Sets a responder to the default responder.

`#(config ssl ocsp)`**`delete`** *`responder_name`*
Deletes the specified responder.

`#(config ssl ocsp)` **`exit`**
Exits the `config ssl ocsp` prompt and returns to the `config ssl` prompt.

`#(config ssl ocsp)`**`no`**
Clears the current default responder setting.

`#(config ssl ocsp)`**`view`**
Displays configuration information for each responder.

`#(config ssl ocsp)`**`edit`** *`responder_name`*
Configure this *`responder_name`*.

Changes the prompt to:

`#(config ocsp` *`responder_name`*`)`

`#(config ocsp` *`responder_name`*`)`**`exit`**
Exits the `config ocsp` *`responder_name`* prompt and returns to the `config ssl ocsp` prompt.

`#(config ocsp` *`responder_name`*`)`**`extension nonce {disable | enable}`**
Enables or disables use of a nonce control in an OCSP request. When enabled, a nonce (unique digits sequence) is included as one of the requestExtensions in each OCSP request. Default is disable.

`#(config ocsp` *`responder_name`*`)`**`extension request-signing-keyring`** *`keyring-id`*
Configures the OCSP request to contain a signature along with certificates to help the OCSP responder verify this signature. They keyring must already exist and have a certificate.

`#(config ocsp` *`responder_name`*`)`**`ignore expired-responder {enable | disable}`**
Specifies whether the OCSP request must contain a signature along with certificates to help the OCSP responder verify this signature. They keyring must already exist and ave a certificate. By default, invalid responder certificate dates cause the subject certificate verification to fail.

`#(config ocsp` *`responder_name`*`)`**`ignore ocsp-signing-purpose {enable | disable}`**

Specifies whether to ignore the enforcement of purpose field in the responder certificate. Default is enable.

`#(config ocsp` *`responder_name`*`)`**`ignore request-failure {enable | disable}`**
Specifies whether to ignore connection failures and timeouts to the OCSP server. Default is disable.

`#(config ocsp` *`responder_name`*`)`**`ignore unknown-status {enable | disable}`**
Specifies whether to treat "unknown" revocation status for a certificate as an error. By default, unknown status is an error and causes subject certification verification to fail.

`#(config ocsp` *`responder_name`*`)`**`ignore untrusted-responder {enable | disable}`**
Specifies whether to bypass, during responder certificate verification, any untrusted certificate errors. For example, a missing issuer certificate or a missing self-signed certificate. By default, any untrusted certificate failure is an error and causes the subject certificate verification to fail.

`#(config ocsp` *`responder_name`*`)`**`issuer-ccl`** `{`*`CCL Name`*` | all | none}`
Sets the name of the CCL. This is the list of CA names which is associated with the certificate to be checked for revocation. It may either be a server or client certificate, or a certificate that is used for verifying system images.

`#(config ocsp` *`responder_name`*`)`**`no extension request-signing-keyring`**
Resets the request signing keyring.

`#(config ocsp` *`responder_name`*`)`**`response-ccl`** `{`*`Response CCL Name`*` |` **`all`**`}`
Sets the name of the CCL.

`#(config ocsp` *`responder_name`*`)`**`ssl-device-profile`** *`SSL device-profile name`*
Sets the SSL device profile. The device profile is a unique set of SSL cipher-suites, protocols and keyrings used when the ProxySG makes HTTPS connections with an OCSP responder. The default value is the pre-created device profile named "default."

`#(config ocsp` *`responder_name`*`)`**`ttl`** `{`**`auto`**` |` *`number_ of_days`*`}`
Configures the time to live (TTL) value. This value determines how long a response remains in the cache. The auto option indicates that the response is cached until nextUpdate. If nextUpdate is not present the response is not cached. The *`number_ of_days`* variable indicates that the nextUpdate field in the response is to be overridden and that the response is to be cached for the indicated number of days. Default is auto.

`#(config ocsp` *`responder_name`*`)`**`url`** *`ocsp server url`*
Configures the time to live (TTL) value. This value determines how long a response remains in the cache. The auto option indicates that the response is cached until nextUpdate. If nextUpdate is not present the response is not cached. The *`number_of_days`* variable indicates that the nextUpdate field in the response is to be overridden and that the response is to be cached for the indicated number of days. Default is auto.

`#(config ocsp` *`responder_name`*`)` **`use-forwarding {disable | enable}`**
Sets the OCSP requests to use forwarding.

`#(config ocsp` *`responder_name`*`)` **`view`**
Displays the responder configurations.

For More Information

- *SGOS Administration Guide*

Example

```
SGOS#(config) ssl
SGOS#(config ssl) edit ssl-client ssl_default_client_name
SGOS#(config ssl ssl-client ssl_default_client_name) cipher-suite rc4-md5
exp-des-cbc-sha
 ok
SGOS#(config ssl ssl-client ssl_default_client_name) exit
SGOS#(config ssl) exit
SGOS#(config)
```

#(config) static-routes

Synopsis

Use this command to set the network path to download the static routes configuration file.

To use static routes on the ProxySG, you must create a routing table and place it on an HTTP server accessible to the device. The routing table is a text file containing a list of static routes made up of destination IP addresses (IPv4 or IPv6), subnet masks (for IPv4) or prefix lengths (for IPv6), and gateway IP addresses (IPv4 or IPv6). When you download a routing table, the table is stored in the device until it is replaced by downloading a new table.

The routing table is a simple text file containing a list of IPv4/IPv6 addresses, subnet masks/prefix lengths, and gateways. A sample routing table is illustrated below:

```
10.63.0.0                    255.255.0.0         10.63.158.213
10.64.0.0                    255.255.0.0         10.63.158.213
10.65.0.0                    255.255.0.0         10.63.158.226
fe80::2d0:83ff:fe05:780%0:0  64                  00:D0:83:05:07:80
```

Note that a routing table can contain a combination of IPv4 and IPv6 entries, but the gateway for each destination must be on the appropriate network type. For example, an IPv6 destination must use an IPv6 gateway.

When a routing table is loaded, all requested addresses are compared to the list, and routed based on the best match.

After the routing table is created, place it on an HTTP server so it can be downloaded to the device. To download the routing table to the ProxySG, use the `load` command.

Syntax

`#(config)` **`static-routes no path`**
Clears the network path location of the static route table

`#(config)` **`static-routes path`** *`url`*
Sets the network path location of the static route table to the specified URL.

For More Information

- *SGOS Administration Guide*

Example

```
SGOS#(config) static-routes path 10.25.36.47/files/routes.txt
    ok
```

#(config) statistics-export

Synopsis

Configure the parameters for exporting statistical data from the ProxySG to an external data collector.

Syntax

```
#(config) statistics-export
```

This changes the prompt to:

```
#(config statistics-export)
```

Subcommands

`#(config statistics-export)` **`config-path`** *URL*
Identifies the location of configuration file on the external data collector.

`#(config statistics-export)` **`force-export`**
Sends the exported statistic data at the start of the next minute. The time remaining until the next data export is displayed to the user in seconds. For example, `Next data export will happen in 8 seconds`.

`#(config statistics-export)` **`reread-config`**
Reads the contents of the configuration file on a remote server, and applies any changes.

`#(config statistics-export)` **`ssl-device-profile`** *device-profile-name*
When using HTTPS, this command identifies which SSL device profile to use when contacting the external data collector.

`#(config statistics-export)` **`disable`**
Disables exporting statistics to the external data collector.

`#(config statistics-export)` **`enable`**
Enables exporting statistics to the external data collector.

`#(config statistics-export)` **`view`**
Shows the settings for exporting statistics.

`#(config statistics-export)` **`exit`**
Exits `#(config statistics-export)` mode and returns to `#(config)` mode.

Example

```
SGOS#(config) statistics-export
SGOS#(config statistics-export) ssl-device-profile data-collector-df
 ok
SGOS#(config statistics-export) config-path http://10.167.0.116/config.txt
 ok
SGOS#(config statistics-export) enable
 ok
```

View the results.

```
#(config statistics-export)view
Statistics export configuration
  Statistics export:                Enabled
  Configuration path:               https://10.167.0.116/config.txt
  SSL device profile:               data-collector-df
  Configuration information:
    Details of last configuration download:
        Configuration path:         http://10.167.0.116/config.txt?version=1-
              1&sn=4417164142&ip=10.167.1.219&model=300-5
        Last attempted config:      2011-09-01 20:14:01 UTC
        Last successful config:     2011-09-01 20:14:01 UTC
    Details of active configuration:
        Version:                    1
        Time interval:              15 minutes
        Trend filter:               http
        Upload path:                https://10.167.0.116/cgi-bin/post.py

  Upload information:
    Details of last upload:
        Upload path:                https://10.167.0.116/cgi-bin/post.py
        Last attempted upload time: 2011-09-01 20:14:01 UTC
        Last successful upload time: 2011-09-01 20:14:01 UTC
    Next estimated upload time:     2011-09-01 20:34:07 UTC
    Successful uploads:             11
    Failed upload attempts:         0
    Data lost in minutes:           0
```

#(config) streaming

Synopsis

Use this command to configure global streaming settings as well as settings for each streaming proxy (Windows Media, Real Media, QuickTime, Flash, Adobe HDS, Apple HLS, and Microsoft Smooth Streaming).

Syntax

`#(config) streaming adobe-hds http-handoff {disable | enable}`

Disables or enables Adobe HDS handoff. Set to enable in order to control and view statistics on Adobe HTTP Dynamic Streaming streams.

`#(config) streaming apple-hls http-handoff {disable | enable}`

Disables or enables Apple HLS handoff. Set to enable in order to control and view statistics on Apple HTTP Live Streaming S streams

`#(config) streaming flash http-handoff {disable | enable}`
Disables or enables Flash HTTP handoff.

`#(config) streaming max-client-bandwidth kbps`
Sets the maximum client bandwidth permitted to *kbps*.

`#(config) streaming max-gateway-bandwidth kbps`
Sets the maximum gateway bandwidth permitted to *kbps*.

`#(config) streaming ms-smooth http-handoff {disable | enable}`

Disables or enables Microsoft Smooth handoff.

`#(config) streaming multicast address-range first_address - last_address`
The IP address range for the ProxySG's multicast-station. Default is from 224.2.128.0 and 224.2.255.255.

`#(config) streaming multicast port-range first_port - last_port`
Port range for the ProxySG's multicast-station. Default is between 32768 and 65535.

`#(config) streaming multicast ttl ttl`
Time to live value for the multicast-station on the ProxySG, expressed in hops. Default is 5; a valid number is between 1 and 255.

`#(config) streaming no max-client-bandwidth`
Clears the current maximum client bandwidth setting.

`#(config) streaming no max-gateway-bandwidth`
Clears the current maximum gateway bandwidth setting.

`#(config) streaming quicktime http-handoff {disable | enable}`
Disables or enables QuickTime HTTP handoff.

`#(config) streaming quicktime max-client-bandwidth kbps`
Sets the maximum connections allowed.

`#(config) streaming quicktime max-connections number`
Sets the maximum client bandwidth allowed.

`#(config) streaming quicktime max-gateway-bandwidth kbps`
Sets the maximum gateway bandwidth allowed.

`#(config) streaming quicktime no {max-client-bandwidth | max-connections | max-gateway-bandwidth}`
Negates QuickTime parameters.

`#(config)` **`streaming real-media http-handoff {disable | enable}`**
Disables or enables Real Media HTTP handoff.

`#(config)` **`streaming real-media log-forwarding {disable | enable}`**
Sets Real Media client log forwarding.

`#(config)` **`streaming real-media max-client-bandwidth`** *`kbps`*
Limits the total bandwidth used by all connected clients. Changing the setting to no max-client-bandwidth uses the maximum available bandwidth. Zero (0) is not an accepted value

`#(config)` **`streaming real-media max-connections`** *`number`*
Limits the concurrent number of client connections. Changing the setting to `no max-connections` uses the maximum available bandwidth. Zero (0) is not an accepted value.

`#(config)` **`streaming real-media max-gateway-bandwidth`** *`kbps`*
Limits the total bandwidth used between the proxy and the gateway. Changing the setting to `no max-gateway-bandwidth`, uses the maximum available bandwidth. Zero (0) is not an accepted value.

`#(config)` **`streaming real-media multicast {disable | enable}`**
Disables or enables Real Media client multicast support.

`#(config)` **`streaming real-media no {max-client-bandwidth | max-connections | max-gateway-bandwidth | refresh-interval}`**
Negates Real Media parameters.

`#(config)` **`streaming real-media refresh-interval`** *`hours`*
Sets the streaming content refresh interval.

`#(config)` **`streaming windows-media asx-rewrite`** `number in_addr cache_proto cache_addr [cache-port]`
Provides proxy support for Windows Player 6.4.

If your environment does not use a Layer 4 switch or WCCP, the ProxySG can operate as a proxy for Windows Media Player 6.4 clients by rewriting the `.asx` file (which links Web pages to Windows Media ASF files) to point to the Windows Media streaming media cache rather than the Windows Media server.

`number` can be any positive number. It defines the priority of all the asx-rewrite rules. Smaller numbers indicate higher priority. *`in_addr`* specifies the hostname. It can have a maximum of one wildcard character. *`cache_proto`* rewrites the protocol on the ProxySG and can take any of the following forms:

`mmsu` (MMS-UDP)

`mmst` (MMS-TCP)

`http` (HTTP)

`mms` (MMS-UDP or MMS-TCP)

`cache_addr` rewrites the address on the ProxySG.

`#(config)` **`streaming windows-media broadcast-alias`** *`alias url loops date time`*
Enables scheduled live unicast or multicast transmission of video-on-demand content.

`alias` must be unique. *`url`* specifies the address of the video-on-demand stream. *`loops`* specifies the number of times the stream should be played back. 0 means forever. *`date`* specifies the broadcast alias starting date. To specify multiple starting dates, enter the date as a comma-separated string. *`date`* can take any of the following formats:

yyyy-mm-dd

today

`time` specifies the broadcast-alias starting time. To specify multiple starting times within the same date, enter the time as a comma-separated string. No spaces are permitted. *time* can take any of the following formats:

`hh:mm`

`midnight, 12am, 1am, 2am, 3am, 4am, 5am, 6am, 7am, 8am, 9am, 10am, 11am, noon, 12pm, 1pm, 2pm, 3pm, 4pm, 5pm, 6pm, 7pm, 8pm, 9pm, 10pm, 11pm`.

`#(config)` **`streaming windows-media http-handoff {disable | enable}`**
Allows the Windows Media module to control the HTTP port when Windows Media streaming content is present. The default is enabled.

`#(config)` **`streaming windows-media live-retransmit {disable | enable}`**
Allows the ProxySG to retransmit dropped packets sent through MMS-UDP for unicast. The default is enabled.

`#(config)` **`streaming windows-media log-compatibility {disable | enable}`**
Disables or enables access log compatibility. When log-compatibility is enabled, the ProxySG generates the MMS log the same way as Windows Media Server does. Three fields are affected when log-compatibility is enabled:

c-ip x-wm-c-ip (client address derived from client log)

c-dns x-wm-c-dns (client hostname derived from client log)

c-uri-stem cs-uri (use full URI instead of just the path)

`#(config)` **`streaming windows-media log-forwarding {disable | enable}`**
Enables or disables forwarding of the client log to the origin media server.

`#(config)` **`streaming windows-media max-client-bandwidth`** *`kpbs`*
Sets the maximum client bandwidth permitted to *`kbps`*.

`#(config)` **`streaming windows-media max-connections`** *`number`*
Limits the concurrent number of client connections. If this variable is set to 0, you effectively lock out all client connections to the ProxySG. To allow maximum client bandwidth, enter **`streaming windows-media no max-connections`**.

`#(config)` **`streaming windows-media max-fast-bandwidth`** *`kpbs`*
Sets the maximum fast start bandwidth per player.

`#(config)` **`streaming windows-media max-gateway-bandwidth`** *`kpbs`*
Sets the maximum limit, in kilobits per second (Kbps), for the amount of bandwidth Windows Media uses to send requests to its gateway. If this variable is set to 0, you effectively prevent the ProxySG from initiating any connections to the gateway. To allow maximum gateway bandwidth, enter **`streaming windows-media no max-gateway-bandwidth`**.

`#(config)` **`streaming windows-media multicast-alias`** *`alias url`* `[preload]`
Creates an alias on the ProxySG that reflects the multicast station on the origin content server.

`#(config)` **`streaming windows-media multicast-error-correction {disable | enable}`**
Enables the transmission of forward error correction (FEC) packets from a Windows Media Server to proxied Windows Media Player clients, when provided. This feature is enabled by default. Disabling the option can lead to a bandwidth gain but may lead to playback instability on high-latency Internet connections.

`#(config)` **`streaming windows-media multicast-station`** *`name`* `{`*`alias`*` | `*`url`*`}` *`ip port ttl`*
Enables multicast transmission of Windows Media content from the ProxySG. *`name`* specifies the name of the alias. It must be unique. *`alias`* can be a unicast alias, a multicast-alias or a broadcast alias, as well as a *`url`* to a live stream source. *`ip`* is an optional parameter and specifies the multicast station's IP address. *`port`* specifies the multicast station's port value address. *`ttl`* specifies the multicast-station's time-to-live value, expressed in hops (and must be a valid number between 1 and 255). The default *`ttl`* is 5.

`#(config)` **`streaming windows-media no asx-rewrite`** *`number`*
Deletes the ASX rewrite rule associated with *`number`*.

`#(config)` **`streaming windows-media no broadcast-alias`** *`alias`*
Deletes the broadcast alias rule associated with *`alias`*.

`#(config)` **`streaming windows-media no max-client-bandwidth`**
Negates maximum client bandwidth settings.

`#(config)` **`streaming windows-media no max-connections`**
Negates maximum connections settings.

`#(config)` **`streaming windows-media no max-gateway-bandwidth`**
Negates maximum gateway bandwidth settings.

`#(config)` **`streaming windows-media no multicast-alias`** *`alias`*
Deletes the multicast alias rule associated with *`alias`*.

`#(config)` **`streaming windows-media no multicast-station`** *`name`*
Deletes the multicast station rule associated with *`name`*.

`#(config)` **`streaming windows-media no refresh-interval`**
Sets the current Windows Media refresh interval to "never refresh."

`#(config)` **`streaming windows-media no server-auth-type`** *`cache_ip_address`*
Clears the authentication type associated with *`cache_ip_address`*.

`#(config)` **`streaming windows-media no unicast-alias`** *`alias`*
Deletes the unicast alias rule associated with *`alias`*. The name of the alias, such as "welcome1" that is created on the ProxySG and reflects the content specified by the URL. The protocol is specified by the URL if the protocol is `mmst`, `mmsu`, or `http`. If the protocol is `mms`, the same protocol as the client is used.

`#(config)` **`streaming windows-media refresh-interval`** *`hours`*
Checks the refresh interval for cached streaming content. *`hours`* must be a floating point number to specify refresh interval. 0 means always check for freshness.

`#(config)` **`streaming windows-media server-auth-type`** {**`basic`** | **`ntlm`**} *`cache_ip_address`*
Sets the authentication type of the ProxySG indicated by *`cache_ip_address`* to BASIC or NTLM.

`#(config)` **`streaming windows-media server-thinning {disable | enable}`**
Disables or enables server thinning.

`#(config)` **`streaming windows-media unicast-alias`** *`alias url`*
Creates an alias on the ProxySG that reflects the content specified by the URL. When a client requests the alias content, the ProxySG uses the URL specified in the `unicast-alias` command to request the content from the origin streaming server.

For More Information

- *SGOS Administration Guide*, Streaming chapter

Example

```
SGOS#(config) streaming windows-media http-handoff enable
 ok
SGOS#(config) streaming windows-media live-retransmit disable
 ok
SGOS#(config) streaming windows-media log-forwarding disable
 ok
SGOS#(config) streaming windows-media max-connections 1600
 ok
SGOS#(config) streaming windows-media no max-connections
 ok
```

#(config) tcp-ip

Synopsis

Use the following commands to configure your TCP-IP settings.

Syntax

`#(config)` **`tcp-ip bypass-keep-alive {disable | enable}`**
Enables or disables keep-alive for bypass connections. Note that this setting does not affect existing connections.

`#(config)` **`tcp-ip icmp-bcast-echo {disable | enable}`**
Enables or disables ICMP broadcast echo responses.

`#(config)` **`tcp-ip icmp-tstamp-echo {disable | enable}`**
Enables or disables ICMP timestamp echo responses.

`#(config)` **`tcp-ip ip-forwarding {disable | enable}`**
Enables or disables IP-forwarding.

`#(config)` **`tcp-ip pmtu-discovery {disable | enable}`**
Enables or disables Path MTU Discovery.

`#(config)` **`tcp-ip rfc-1323 {disable | enable}`**
Enables or disables RFC-1323 support (satellite communications).

`#(config)` **`tcp-ip routing-algorithm hashing [both | destination-address | source-address]`**
Sets the routing algorithm as hashing. Selects the outbound route within the same group based on source and/or destination IP address. Use the hashing option, for example, when the ProxySG appliance needs to connect to a secure Web server and the Web server requires the source IP address to remain unchanged during the lifetime of the secure session. Similarly, other services that use cookies to maintain session concept across multiple connections might also need to use hash base routing. The default setting is `weighted-round-robin`.

`#(config)` **`tcp-ip routing-algorithm weighted-round-robin`**
The default setting for the `tcp-ip routing-algorithm` option is `weighted-round-robin`. This setting is appropriate for all deployments except where noted in the `hashing` above.

`#(config)` **`tcp-ip tcp-newreno {disable | enable}`**
Enables or disables TCP NewReno support (improved fast recovery).

`#(config)` **`tcp-ip tcp-2msl`** *`seconds`*
Specifies the `time_wait` value for a TCP connection before completely closing.

`#(config)` **`tcp-ip tcp-loss-recovery-mode {aggressive | enhanced | normal}`**
Helps to recover throughput efficiently after packet losses occur and also addresses performance problems due to a single packet loss during a large transfer over long delay pipes. The feature is enabled (set to `normal`) by default.

`#(config)` **`tcp-ip window-size`** *`window_size`*
Specifies the TCP window size for satellite communications.

Example

```
SGOS#(config) tcp-ip ip-forwarding enable
 ok
SGOS#(config) tcp-ip rfc-1323 enable
 ok
```

#(config) tcp-ip scps

Synopsis

Use the following commands to configure your TCP-IP SCPS settings.

Syntax

`#(config)` **`tcp-ip scps`** {**`disable`** | **`enable`**}
Enables or disables SCPS-TP protocol.

`#(config)` **`tcp-ip scps bandwidth`** *`bandwidth_size`*
Specifies the transmission link bandwidth to be used by the ProxySG appliance for packet metering and window sizing during SCPS usage. The value is kbps (shown as bits-per-second/1000).

`#(config)` **`tcp-ip scps interface`** *`adapter:interface[.vlan]`*
Sets the satellite-facing interface used to communicate with the satellite during a SCPS transmission.

`#(config)` **`tcp-ip scps rtt`** *`rtt_value`*
Sets the SCPS link round-trip time. The value is in milliseconds.

For More Information

- *SCPS Deployment Guide*

Example

```
SGOS#(config) tcp-ip scps rtt 570
 ok
SGOS#(config) (config)tcp-ip scps bandwidth 1544
 ok#
SGOS#(config) (config)tcp-ip scps interface 0:0
 ok
```

#(config) threat-protection

Synopsis

Use the following commands to configure threat-protection in your network. These commands set the defaults for the built-in threat protection policy that is invoked when you enable malware scanning on the ProxySG. When malware scanning is enabled, the ProxySG and the Blue Coat AV work in conjunction to analyze incoming Web content and apply policy protect users from malware and malicious content.

Syntax

- To enter configuration mode:

```
SGOS#(config) threat-protection
```

This changes the prompt to:

```
SGOS#(config threat-protection) [subcommands]
```

- The following subcommands are available:

```
SGOS# (config threat-protection)exit
```

Allows you to exit from the threat-protection configuration to the configuration prompt.

```
SGOS# (config threat-protection) view
```

Allows you to view the threat protection settings.

```
SGOS# (config threat-protection) malware-scanning
```

Allows you to configure the malware-scanning parameters that will be compiled in the built-in threat protection policy file.

```
SGOS# (config threat-protection malware-scanning) {disable | enable | exit}
```

Allows you to disable, enable, or exit the malware scanning configuration options.

```
SGOS# (config threat-protection malware-scanning) failure-mode {continue | deny}
```

Allows you to set the action on an unsuccessful scan on the ProxySG. Continue allows the ProxySG to serve the content even if the Blue Coat AV was unable to complete the scan of the requested Web content.

If set to deny, the ProxySG will not serve the requested Web content to the user, in the event that the Blue Coat AV is unable to complete the scan.

```
SGOS# (config threat-protection malware-scanning) level {high-performance | maximum-protection}
```

Allows you to implement the network performance rules or the network protection rules based on your preferences in the malware scanning configuration.

The threat protection policy offers two levels for scanning responses redirected to the Blue Coat AV — high performance and maximum security. While the Blue Coat AV scans all Web responses when set to maximum security, it selectively scans Web responses when set to high performance bypassing content that has a low risk of malware infection.

```
SGOS# (config threat-protection malware-scanning) no update-path
```

Clears the update path URL that the ProxySG uses to obtain the latest malware threat-protection policy file.

`SGOS# (config threat-protection malware-scanning) secure-connection {always | if-available | never}`

The communication between the ProxySG and the Blue Coat AV can be in plain ICAP, secure ICAP or can use both plain and secure ICAP, depending on whether the response processed by the ProxySG uses the HTTP, FTP, or HTTPS protocol.

This option allows you to configure whether a secure connection is used always, if-available, or is never used.

`SGOS# (config threat-protection malware-scanning) update-path <url>`

Provides the path to the URL where updates to the threat protection solution are posted. Updates to the threat protection solution are available as a gzipped tar archive file which can be downloaded to a local Web server in your network or installed directly on the ProxySG

`SGOS# (config threat-protection malware-scanning) view}`

Displays the configuration of the malware scanning policy that is currently implemented on the ProxySG.

Related Commands

`SGOS#(config) show sources policy threat-protection`

Displays the source file for the threat-protection policy.

`SGOS#(config) load threat-protection malware-scanning`

Downloads the updates to the malware scanning rules included in the threat-protection policy file.

Example

To view the malware scanning configuration on the ProxySG:

```
SGOS#(config) threat-protection
SGOS#(config threat-protection) malware-scanning
SGOS#(config threat-protection malware-scanning)view
Malware scanning solution: enabled
Threat protection level: high-performance
Secure scanner connection: if-available
Failure mode: deny
Update URL:
```

To download and install the latest threat-protection policy file:

```
SGOS#(config) threat-protection
SGOS#(config threat-protection) malware-scanning
SGOS#(config threat-protection malware-scanning)update-path
https://bto.bluecoat.com/download/modules/security/SGv6/threatprotection.tar.gz
ok
SGOS#(config threat-protection malware-scanning)exit
SGOS#(config threat-protection)exit
SGOS#(config)load threat-protection malware-scanning
```

#(config) timezone

Synopsis

Use this command to set the local time zone on the ProxySG.

Syntax

`#(config)` **`timezone set`** *`area/location`*
Enables you to set the local time zone. (Use `(config) show timezones` to display a list of supported timezones.)

`#(config)` **`timezone database-path`** *`url`* `|` **`default`**
Sets the network path to download the Time zone database.

For More Information

- *SGOS Administration Guide*
- **`#(config) clock`** on page 149

Example

```
SGOS#(config) timezone 3
 ok
```

#(config) ui

Synopsis

Use this command to configure the UI settings for the ProxySG.

Syntax

```
#(config) ui
```

This changes the prompt to:

```
#(config ui)
```

Subcommands

`#(config ui)` **`default {advanced | solution}`**
Sets the default user interface.

`#(config ui)` **`exit`**
Exits UI mode and returns to the `#(config)` prompt.

`#(config ui)` **`no update-path`**
Clears the new UI download path.

`#(config ui)` **`reset`**
Resets the UI to the bound system version.

`#(config ui)` **`update-path`** *`url`*
Sets the new UI download path.

For More Information

- *SGOS Administration Guide*

Example

```
#(config ui) default advanced
 ok
```

#(config) upgrade-path

Synopsis

Use this command to specify the network path to download system software.

Syntax

`#(config)` **`upgrade-path`** *`url`*

Indicates the network path to use to download ProxySG system software. The image name must be included in the network path.

Example

```
SGOS#(config) upgrade-path http://your_server/ProxySG_5.3.1.9_36410_200.CHK
ok
```

#(config) virtual-ip

Synopsis

This command allows you to configure virtual IP addresses.

Syntax

`#(config)` **`virtual-ip address`** *`ip_address`*
Specifies the virtual IP to add.

`#(config)` **`virtual-ip clear`**
Removes all virtual IP addresses.

`#(config)` **`virtual-ip no address`** *`ip_address`*
Removes the specified virtual IP from the list.

For More Information

- *SGOS Administration Guide*
- **`#(config) failover`** on page 198

Example

```
SGOS#(config) virtual-ip address 10.25.36.47
 ok
```

#(config) wccp

Synopsis

The ProxySG can be configured to participate in a WCCP (Web Cache Control Protocol) scheme, where a WCCP-capable router collaborates with a set of WCCP-configured ProxySG to service requests. WCCP is a Cisco-developed protocol. For more information about WCCP, refer to the WCCP chapter in the *SGOS Administration Guide* and the *WCCP Deployment Guide*.

After you have created the WCCP configuration file, place the file on an HTTP server so it can be downloaded to the ProxySG. To download the WCCP configuration to the ProxySG, use the `load` command.

Syntax

`#(config)` **`wccp disable`**
Disables WCCP.

`#(config)` **`wccp enable`**
Enables WCCP.

`#(config)` **`wccp no path`**
Negates certain WCCP settings.

`#(config)` **`wccp path`** *`url`*
Specifies the network path from which to download WCCP settings.

For More Information

- *SGOS Administration Guide*

Example

```
SGOS#(config) wccp path 10.25.36.47/files/wccp.txt
 ok
```