



Using the Command-Line Interface

This describes how to use the CLI on the security appliance, and includes the following topics:

- [Firewall Mode and Security Context Mode, page 1-1](#)
- [Command Modes and Prompts, page 1-2](#)
- [Syntax Formatting, page 1-3](#)
- [Abbreviating Commands, page 1-3](#)
- [Command-Line Editing, page 1-3](#)
- [Command Completion, page 1-3](#)
- [Command Help, page 1-4](#)
- [Filtering show Command Output, page 1-4](#)
- [Command Output Paging, page 1-5](#)
- [Adding Comments, page 1-5](#)
- [Text Configuration Files, page 1-6](#)



Note

The CLI uses similar syntax and other conventions to the Cisco IOS CLI, but the security appliance operating system is not a version of Cisco IOS software. Do not assume that a Cisco IOS CLI command works with or has the same function on the security appliance.

Firewall Mode and Security Context Mode

The security appliance runs in a combination of the following modes:

- Transparent firewall or routed firewall mode

The firewall mode determines if the security appliance runs as a Layer 2 or Layer 3 firewall.

- Multiple context or single context mode

The security context mode determines if the security appliance runs as a single device or as multiple security contexts, which act like virtual devices.

Some commands are only available in certain modes.

Command Modes and Prompts

The security appliance CLI includes command modes. Some commands can only be entered in certain modes. For example, to enter commands that show sensitive information, you need to enter a password and enter a more privileged mode. Then, to ensure that configuration changes are not entered accidentally, you have to enter a configuration mode. All lower commands can be entered in higher modes, for example, you can enter a privileged EXEC command in global configuration mode.

When you are in the system configuration or in single context mode, the prompt begins with the hostname:

```
hostname
```

When you are within a context, the prompt begins with the hostname followed by the context name:

```
hostname/context
```

The prompt changes depending on the access mode:

- User EXEC mode

User EXEC mode lets you see minimum security appliance settings. The user EXEC mode prompt appears as follows when you first access the security appliance:

```
hostname>
```

```
hostname/context>
```

- Privileged EXEC mode

Privileged EXEC mode lets you see all current settings up to your privilege level. Any user EXEC mode command will work in privileged EXEC mode. Enter the **enable** command in user EXEC mode, which requires a password, to start privileged EXEC mode. The prompt includes the number sign (#):

```
hostname#
```

```
hostname/context#
```

- Global configuration mode

Global configuration mode lets you change the security appliance configuration. All user EXEC, privileged EXEC, and global configuration commands are available in this mode. Enter the **configure terminal** command in privileged EXEC mode to start global configuration mode. The prompt changes to the following:

```
hostname(config)#
```

```
hostname/context(config)#
```

- Command-specific configuration modes

From global configuration mode, some commands enter a command-specific configuration mode. All user EXEC, privileged EXEC, global configuration, and command-specific configuration commands are available in this mode. For example, the **interface** command enters interface configuration mode. The prompt changes to the following:

```
hostname(config-if)#
```

```
hostname/context(config-if)#
```

Syntax Formatting

Command syntax descriptions use the following conventions:

Table 1-1 **Syntax Conventions**

Convention	Description
bold	Bold text indicates commands and keywords that you enter literally as shown.
<i>italics</i>	Italic text indicates arguments for which you supply values.
[x]	Square brackets enclose an optional element (keyword or argument).
	A vertical bar indicates a choice within an optional or required set of keywords or arguments.
[x y]	Square brackets enclosing keywords or arguments separated by a vertical bar indicate an optional choice.
{x y}	Braces enclosing keywords or arguments separated by a vertical bar indicate a required choice.
[x {y z}]	Nested sets of square brackets or braces indicate optional or required choices within optional or required elements. Braces and a vertical bar within square brackets indicate a required choice within an optional element.

Abbreviating Commands

You can abbreviate most commands down to the fewest unique characters for a command; for example, you can enter **wr t** to view the configuration instead of entering the full command **write terminal**, or you can enter **en** to start privileged mode and **conf t** to start configuration mode. In addition, you can enter **0** to represent **0.0.0.0**.

Command-Line Editing

The security appliance uses the same command-line editing conventions as Cisco IOS software. You can view all previously entered commands with the **show history** command or individually with the up arrow or **^p** command. Once you have examined a previously entered command, you can move forward in the list with the down arrow or **^n** command. When you reach a command you wish to reuse, you can edit it or press the **Enter** key to start it. You can also delete the word to the left of the cursor with **^w**, or erase the line with **^u**.

The security appliance permits up to 512 characters in a command; additional characters are ignored.

Command Completion

To complete a command or keyword after entering a partial string, press the **Tab** key. The security appliance only completes the command or keyword if the partial string matches only one command or keyword. For example, if you enter **s** and press the **Tab** key, the security appliance does not complete the command because it matches more than one command. However, if you enter **dis**, the **Tab** key completes the command **disable**.

Command Help

Help information is available from the command line by entering the following commands:

- **help** *command_name*
Shows help for the specific command.
- **help** ?
Shows commands for which there is help.
- *command_name* ?
Shows a list of arguments available.
- *string*? (no space)
Lists the possible commands that start with the string.
- ? and +?
Lists all commands available. If you enter ?, the security appliance shows only commands available for the current mode. To show all commands available, including those for lower modes, enter +?.

**Note**

If you want to include a question mark (?) in a command string, you must press **Ctrl-V** before typing the question mark so you do not inadvertently invoke CLI help.

Filtering show Command Output

You can use the vertical bar (|) with any **show** command and include a filter option and filtering expression. The filtering is performed by matching each output line with a regular expression, similar to Cisco IOS software. By selecting different filter options you can include or exclude all output that matches the expression. You can also display all output beginning with the line that matches the expression.

The syntax for using filtering options with the **show** command is as follows:

```
hostname# show command | {include | exclude | begin | grep [-v]} regex
```

In this command string, the first vertical bar (|) is the operator and must be included in the command. This operator directs the output of the **show** command to the filter. In the syntax diagram, the other vertical bars (|) indicate alternative options and are not part of the command.

The **include** option includes all output lines that match the regular expression. The **grep** option without **-v** has the same effect. The **exclude** option excludes all output lines that match the regular expression. The **grep** option with **-v** has the same effect. The **begin** option shows all the output lines starting with the line that matches the regular expression.

Replace *regex* with any Cisco IOS regular expression. See The regular expression is not enclosed in quotes or double-quotes, so be careful with trailing white spaces, which will be taken as part of the regular expression.

When creating regular expressions, you can use any letter or number that you want to match. In addition, certain keyboard characters have special meaning when used in regular expressions. [Table 1-2](#) lists the keyboard characters that have special meaning.

Table 1-2 *Using Special Characters in Regular Expressions*

Character Type	Character	Special Meaning
period	.	Matches any single character, including white space.
asterisk	*	Matches 0 or more sequences of the pattern.
plus sign	+	Matches 1 or more sequences of the pattern.
question mark	? ¹	Matches 0 or 1 occurrences of the pattern.
caret	^	Matches the beginning of the input string.
dollar sign	\$	Matches the end of the input string.
underscore	_	Matches a comma (,), left brace ({), right brace (}), left parenthesis, right parenthesis, the beginning of the input string, the end of the input string, or a space.
brackets	[]	Designates a range of single-character patterns.
hyphen	-	Separates the end points of a range.

1. Precede the question mark with **Ctrl-V** to prevent the question mark from being interpreted as a help command.

To use these special characters as single-character patterns, remove the special meaning by preceding each character with a backslash (\).

Command Output Paging

On commands such as **help** or **?**, **show**, **show xlate**, or other commands that provide long listings, you can determine if the information displays a screen and pauses, or lets the command run to completion. The **pager** command lets you choose the number of lines to display before the More prompt appears.

When paging is enabled, the following prompt appears:

```
<--- More --->
```

The More prompt uses syntax similar to the UNIX **more** command:

- To view another screen, press the Space bar.
- To view the next line, press the **Enter** key.
- To return to the command line, press the **q** key.

Adding Comments

You can precede a line with a colon (:) to create a comment. However, the comment only appears in the command history buffer and not in the configuration. Therefore, you can view the comment with the **show history** command or by pressing an arrow key to retrieve a previous command, but because the comment is not in the configuration, the **write terminal** command does not display it.

Text Configuration Files

This section describes how to format a text configuration file that you can download to the security appliance, and includes the following topics:

- [How Commands Correspond with Lines in the Text File, page 1-6](#)
- [Command-Specific Configuration Mode Commands, page 1-6](#)
- [Automatic Text Entries, page 1-6](#)
- [Line Order, page 1-7](#)
- [Commands Not Included in the Text Configuration, page 1-7](#)
- [Passwords, page 1-7](#)
- [Multiple Security Context Files, page 1-7](#)

How Commands Correspond with Lines in the Text File

The text configuration file includes lines that correspond with the commands described in this guide.

In examples, commands are preceded by a CLI prompt. The prompt in the following example is “hostname(config)#”:

```
hostname(config)# context a
```

In the text configuration file you are not prompted to enter commands, so the prompt is omitted:

```
context a
```

Command-Specific Configuration Mode Commands

Command-specific configuration mode commands appear indented under the main command when entered at the command line. Your text file lines do not need to be indented, as long as the commands appear directly following the main command. For example, the following unindented text is read the same as indented text:

```
interface gigabitethernet0/0
nameif inside
interface gigabitethernet0/1
    nameif outside
```

Automatic Text Entries

When you download a configuration to the security appliance, the security appliance inserts some lines automatically. For example, the security appliance inserts lines for default settings or for the time the configuration was modified. You do not need to enter these automatic entries when you create your text file.

Line Order

For the most part, commands can be in any order in the file. However, some lines, such as ACEs, are processed in the order they appear, and the order can affect the function of the access list. Other commands might also have order requirements. For example, you must enter the **nameif** command for an interface first because many subsequent commands use the name of the interface. Also, commands in a command-specific configuration mode must directly follow the main command.

Commands Not Included in the Text Configuration

Some commands do not insert lines in the configuration. For example, a runtime command such as **show running-config** does not have a corresponding line in the text file.

Passwords

The login, enable, and user passwords are automatically encrypted before they are stored in the configuration. For example, the encrypted form of the password “cisco” might look like jMorNbK0514fadBh. You can copy the configuration passwords to another security appliance in their encrypted form, but you cannot unencrypt the passwords yourself.

If you enter an unencrypted password in a text file, the security appliance does not automatically encrypt them when you copy the configuration to the security appliance. The security appliance only encrypts them when you save the running configuration from the command line using the **copy running-config startup-config** or **write memory** command.

Multiple Security Context Files

For multiple security contexts, the entire configuration consists of multiple parts:

- The security context configurations
- The system configuration, which identifies basic settings for the security appliance, including a list of contexts
- The admin context, which provides network interfaces for the system configuration

The system configuration does not include any interfaces or network settings for itself. Rather, when the system needs to access network resources (such as downloading the contexts from the server), it uses a context that is designated as the admin context.

Each context is similar to a single context mode configuration. The system configuration differs from a context configuration in that the system configuration includes system-only commands (such as a list of all contexts) while other typical commands are not present (such as many interface parameters).

