

Using the Command-Line Interface

This chapter describes how to use the CLI on the Secure Firewall ASA and includes the following sections:

Note The CLI uses similar syntax and other conventions to the Cisco IOS CLI, but the ASA 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 ASA.

- Firewall Mode and Security Context Mode, on page 2
- Command Modes and Prompts, on page 3
- Syntax Formatting, on page 5
- Abbreviating Commands, on page 6
- Command-Line Editing, on page 7
- Command Completion, on page 8
- Command Help, on page 9
- Viewing the Running Configuration, on page 10
- Filtering show and more Command Output, on page 11
- Redirecting and Appending show Command Output , on page 12
- Getting a Line Count for show Command Output, on page 13
- Command Output Paging, on page 14
- Entering Numeric Values, on page 15
- Adding Comments, on page 16
- Text Configuration Files, on page 17
- Supported Character Sets, on page 19

Firewall Mode and Security Context Mode

The ASA runs in a combination of the following modes:

- Transparent firewall or routed firewall mode

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

- Multiple context or single context mode

The security context mode determines if the ASA 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 ASA 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.



Note The various types of prompts are all default prompts and when configured, they can be different.

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

```
ciscoasa
```

- When printing the prompt string, the prompt configuration is parsed and the configured keyword values are printed in the order in which you have set the prompt command. The keyword arguments can be any of the following and in any order: hostname, domain, context, priority, state.

```
asa(config)# prompt hostname context priority state
```

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

```
ciscoasa/context
```

The prompt changes depending on the access mode:

- User EXEC mode

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

```
ciscoasa>
ciscoasa/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 (#):

```
ciscoasa#
ciscoasa/context#
```

- Global configuration mode

Global configuration mode lets you change the ASA 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:

```
ciscoasa(config)#
ciscoasa/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:

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

Syntax Formatting

Command syntax descriptions use the conventions listed in Table 7-1.

Table 1: Syntax Conventions

Convention	Description
bold	Bold text indicates commands and keywords that you enter literally as shown.
italics	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 ASA 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 ASA 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 ASA 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 ASA does not complete the command because it matches more than one command. However, if you enter **dis**, the **Tab** key completes the **disable** command.

Command Help

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

- **help** *command_name*

Shows help for the specific command.

- *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 ASA 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 that you do not inadvertently invoke CLI help.

Viewing the Running Configuration

To view the running configuration, use one of the following commands.

To filter the command output, see the "Filtering show and more Command Output" section on page 7-5.

Command	Purpose
show running-config[**all**][*command*]	Shows the running configuration. If you specify **all**, then all default settings are shown as well. If you specify a *command* , then the output only includes related commands. **Note** Many passwords are shown as *****. To view the passwords in plain text, or in encrypted form if you have a master passphrase enabled, use the **more** command below.
more system:running-config	Shows the running configuration. Passwords are shown in plain text or in encrypted form if you have a master passphrase enabled.

Filtering show and more 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:

```
ciscoasa# show command
 | {include | exclude | begin | grep [-v]} regexp
```

or

```
ciscoasa# more system:running-config
 | {include | exclude | begin | grep [-v]} regexp
```



Note The **more** command can view the contents of any file, not just the running configuration; see the command reference for more information.

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 regexp with any Cisco IOS regular expression. 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 called *metacharacters* have special meaning when used in regular expressions.

Use **Ctrl+V** to escape all of the special characters in the CLI, such as a question mark (?) or a tab. For example, type **d[Ctrl+V]?g** to enter **d?g** in the configuration.

Redirecting and Appending show Command Output

Instead of displaying the output of a **show** command on the screen, you can redirect it to a file on the device or in a remote location. When redirecting to a file on the device, you can also append the command output to the file.

If there is a lot of output to show, command completion can take a long time. For example, you might think the system is stuck if you try to show a million access control entries, or a very large ASP table.

show *command* | {**append** | **redirect**} *url*

- **append** *url* adds the output to an existing file. Specify the file using one of the following:
  - **disk0:/**[[*path*/]*filename*] or **flash:/**[[*path*/]*filename*]—Both **flash** and **disk0** indicate the internal Flash memory. You can use either option.
  - **disk1:/**[[*path*/]*filename*]—Indicates external memory.
- **redirect** *url* creates the specified file, or overwrites it if the file already exists.
  - **disk0:/**[[*path*/]*filename*] or **flash:/**[[*path*/]*filename*]—Both **flash** and **disk0** indicate the internal Flash memory. You can use either option.
  - **disk1:/**[[*path*/]*filename*]—Indicates external memory.
  - **smb:**/[[*path*/]*filename*]—Indicates Server Message Block, a UNIX server local file system.
  - **ftp://**[[*user*[**:***password*]**@**]*server*[*:port*]/[*path*/]*filename*[**;type=***xx*]]—Indicates an FTP server. The **type** can be one of these keywords: **ap** (ASCII passive mode), **an** (ASCII normal mode), **ip** (Default—Binary passive mode), **in** (Binary normal mode).
  - **scp://**[[*user* [**:***password*]**@**]*server* [*/path*]/*filename* [**;int=***interface_name*]]—Indicates an SCP server. The **;int=***interface* option bypasses the route lookup and always uses the specified interface to reach the Secure Copy (SCP) server.
  - **tftp://**[[*user*[**:***password*]**@**]*server*[**:***port*]/[*path*/]*filename*[**;int=***interface_name*]]—Indicates a TFTP server.

Getting a Line Count for show Command Output

Instead of seeing actual **show** command output, you might simply want a count of the number of lines in the output, or the number of lines that match a regular expression. You can then easily compare the line count with the count from previous times you entered the command. This can be a quick check as you make configuration changes. You can either use the **count** keyword, or add **-c** to the **grep** keyword.

show *command* | **count** [*regular_expression*]

show *command* | **grep -c** [*regular_expression*]

Replace regular_expression with any Cisco IOS regular expression. 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. The regular expression is optional; if you do not include one, the count returns the total number of lines in the unfiltered output.

When creating regular expressions, you can use any letter or number that you want to match. In addition, certain keyboard characters called metacharacters have special meaning when used in regular expressions. Use **Ctrl+V** to escape all of the special characters in the CLI, such as a question mark (?) or a tab. For example, type **d[Ctrl+V]?g** to enter **d?g** in the configuration.

For example, to show the total number of all lines in the **show running-config** output:

```
ciscoasa# show running-config | count

Number of lines which match regexp = 271
```

The following example shows how you can quickly check how many interfaces are up. The first example shows how to use the **grep** keyword with a regular expression to filter on only those lines that show an up status. The next example adds the **-c** option to simply show the count rather than the actually lines of output.

```
ciscoasa# show interface | grep is up

Interface GigabitEthernet0/0 "outside", is up, line protocol is up
Interface GigabitEthernet0/1 "inside", is up, line protocol is up

ciscoasa# show interface | grep -c is up

Number of lines which match regexp = 2
```

Command Output Paging

For 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.

Entering Numeric Values

Many commands have parameters that require a numeric value. Unless a non-decimal number is explicitly required, the best practice is to enter the value in decimal. Thus, we provide value ranges in decimal only. For example:

igmp limit *number*

Where *number* is the number of IGMP states allowed on the interface. Valid values range from 0 to 5000. The default value is 500.

You are expected to enter a value in this format within the range, such as:

```
igmp limit 1000
```

Some commands will also take octal or hexadecimal values instead of decimal. How a number is interpreted is based on the following rules:

- Decimal—Numbers like 500 are treated as standard base-10.
- Octal—Numbers prefixed with a zero (0), such as 0500, are interpreted as base-8.
- Hexadecimal—Numbers prefixed with 0x, like 0x500, are interpreted as base-16.

However, if a command does take decimal, octal and hexadecimal numbers are converted to decimal in the running configuration.

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 ASA, and includes the following topics:

- How Commands Correspond with Lines in the Text File
- Command-Specific Configuration Mode Commands
- Automatic Text Entries
- Line Order
- Commands Not Included in the Text Configuration
- Passwords
- Multiple Security Context Files

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 "ciscoasa(config)#":

```
ciscoasa(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 ASA, it inserts some lines automatically. For example, the ASA 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 ASA in its encrypted form, but you cannot unencrypt the passwords yourself.

If you enter an unencrypted password in a text file, the ASA does not automatically encrypt it when you copy the configuration to the ASA. The ASA only encrypts it 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 the following multiple parts:

- The security context configurations
- The system configuration, which identifies basic settings for the ASA, 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).

Supported Character Sets

The ASA CLI currently supports UTF-8 encoding only. UTF-8 is the particular encoding scheme for Unicode symbols, and has been designed to be compatible with an ASCII subset of symbols. ASCII characters are represented in UTF-8 as one-byte characters. All other characters are represented in UTF-8 as multibyte symbols.

The ASCII printable characters (0x20 to 0x7e) are fully supported. The printable ASCII characters are the same as ISO 8859-1. UTF-8 is a superset of ISO 8859-1, so the first 256 characters (0-255) are the same as ISO 8859-1. The ASA CLI supports up to 255 characters (multibyte characters) of ISO 8859-1.