Cisco Intrusion Prevention System Sensor CLI Configuration Guide for IPS 5.1

Americas Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
http://www.cisco.com
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 527-0883

Text Part Number: OL-8677-01

Cisco Intrusion Prevention System Sensor CLI Configuration Guide for IPS 5.1

CONTENTS

Preface

Revised: July 9, 2012, OL-8677-01

Contents

This document describes how to configure the sensor using the IPS 5.1 CLI. It contains the following topics:

Audience

This guide is intended for administrators who need to do the following:

- Configure the sensor for intrusion prevention using the CLI.
- Secure their network with IPS sensors.
- Prevent intrusion on their networks and monitor subsequent alerts.

Conventions

This document uses the following conventions:

Convention	Indication
bold font	Commands and keywords and user-entered text appear in **bold** font.
italic font	Document titles, new or emphasized terms, and arguments for which you supply values are in *italic* font.
[ ]	Elements in square brackets are optional.
{x | y | z }	Required alternative keywords are grouped in braces and separated by vertical bars.

[x | y | z]	Optional alternative keywords are grouped in brackets and separated by vertical bars.
string	A nonquoted set of characters. Do not use quotation marks around the string or the string will include the quotation marks.
`courier` font	Terminal sessions and information the system displays appear in `courier` font.
< >	Nonprinting characters such as passwords are in angle brackets.
[ ]	Default responses to system prompts are in square brackets.
!, #	An exclamation point (!) or a pound sign (#) at the beginning of a line of code indicates a comment line.

Note Means *reader take note*.

Tip Means *the following information will help you solve a problem*.

Caution Means *reader be careful*. In this situation, you might perform an action that could result in equipment damage or loss of data.

Timesaver Means *the described action saves time*. You can save time by performing the action described in the paragraph.

Warning **Means *reader be warned*. In this situation, you might perform an action that could result in bodily injury.**

Related Documentation

For more information on Cisco IPS, refer to the following documentation found at this URL:

http://www.cisco.com/en/US/products/hw/vpndevc/ps4077/tsd_products_support_series_home.html

- *Documentation Roadmap for Cisco Intrusion Prevention System*
- *Release Notes for Cisco Intrusion Prevention System*
- *Cisco Intrusion Prevention System Device Manager Configuration Guide*
- *Cisco Intrusion Prevention System Manager Express Configuration Guide*
- *Cisco Intrusion Prevention System Command Reference*
- *Cisco Intrusion Prevention System Appliance and Module Installation Guide*
- *Installling and Removing Interface Cards in Cisco IPS-4260 and IPS 4270-20*
- *Regulatory Compliance and Safety Information for the Cisco Intrusion Detection and Prevention System 4200 Series Appliance Sensor*

- *Regulatory Compliance and Safety Information for the Cisco ASA 5500-X Series Appliances and the Cisco Intrusion Prevention System 4300 Series Appliances*
- *Regulatory Compliance and Safety Information for the Cisco Intrusion Prevention System 4500 Series Sensor Appliance*

Obtaining Documentation and Submitting a Service Request

For information on obtaining documentation, submitting a service request, and gathering additional information, see the monthly *What's New in Cisco Product Documentation*, which also lists all new and revised Cisco technical documentation, at:

http://www.cisco.com/en/US/docs/general/whatsnew/whatsnew.html

Subscribe to the *What's New in Cisco Product Documentation* as an RSS feed and set content to be delivered directly to your desktop using a reader application. The RSS feeds are a free service. Cisco currently supports RSS Version 2.0.

CHAPTER 1

Introducing the CLI Configuration Guide

This chapter introduces *Configuring the Cisco Intrusion Prevention System Sensor Using the Command Line Interface 5.1*. It contains the following sections:

Overview

This guide is a task-based configuration guide for the IPS 5.1 CLI. The term “sensor” is used throughout this guide to refer to all sensor models, unless a procedure refers specifically to the appliance or one of the modules, such as IDSM-2, NM-CIDS, or AIP-SSM.

For an alphabetical list of all IPS commands, refer to the *Command Reference for Cisco Intrusion Prevention System 5.1*. Refer to the *Documentation Roadmap for Cisco Intrusion Prevention System 5.1* that shipped with your sensor for information on locating all IPS 5.1 documents on Cisco.com.

You can also use an IPS manager to configure your sensor. Refer to the *Documentation Roadmap for Cisco Intrusion Prevention System 5.1* that shipped with your sensor for information on how to access documentation that describes how to use IDM and ASDM.

You can find the IPS 5.1 documentation at this URL:

http://www.cisco.com/en/US/products/hw/vpndevc/ps4077/tsd_products_support_series_home.html

Configuration Sequence

Perform the following tasks to configure the sensor:

1. Log in to the sensor.

   For the procedure, see Chapter 2, "Logging In to the Sensor."

2. Initialize the sensor.

   Run the **setup** command to initialize the sensor.

   For the procedure, see Initializing the Sensor, page 3-2.

3. Verify the sensor initialization.

   For the procedure, see Verifying Initialization, page 3-7.

4. Create the service account.

   A service account is needed for password recovery and other special debug situations directed by TAC.

   For the procedure, see Creating the Service Account, page 4-13.

Caution You should carefully consider whether you want to create a service account. The service account provides shell access to the system, which makes the system vulnerable. However, you can use the service account to create a new password if the Administrator password is lost. Analyze your situation to decide if you want a service account existing on the system.

5. License the sensor.

   For the procedure, see Installing the License Key, page 4-39.

6. Perform the other initial tasks, such as adding users and trusted hosts and so forth.

   For the procedures, see Chapter 4, "Initial Configuration Tasks."

7. Make changes to the interface configuration if necessary.

   For the procedures, see Chapter 5, "Configuring Interfaces."

   Note You configure the interfaces during initialization.

8. Configure event action rules.

   For the procedures, see Chapter 6, "Configuring Event Action Rules."

9. Configure the signatures for intrusion prevention.

   For the procedures, see Chapter 7, "Defining Signatures."

10. Configuring IP Logging.

    For the procedure, see Chapter 8, "Configuring IP Logging."

11. Configure blocking.

    For the procedures, see Chapter 10, "Configuring Attack Response Controller for Blocking and Rate Limiting."

12. Configure SNMP if you are going to use it.

    For the procedures, see Chapter 11, "Configuring SNMP."

13. Perform miscellaneous tasks to keep your sensor running smoothly.

    For the procedures, see Chapter 13, "Administrative Tasks for the Sensor."

14. Upgrade the IPS software with new signature updates and service packs.

    For more information, see Chapter 18, "Obtaining Software."

15. Reimage the application partition and the maintenance partition when needed.

    For the procedures, see Chapter 17, "Upgrading, Downgrading, and Installing System Images."

For procedures specific to the modules, see the following chapters:

- Chapter 14, "Configuring AIP-SSM"
- Chapter 15, "Configuring IDSM-2"
- Chapter 16, "Configuring NM-CIDS"

User Roles

The CLI for IPS 5.1 permits multiple users to log in at a time. You can create and remove users from the local sensor. You can only modify one user account at a time. Each user is associated with a role that controls what that user can and cannot modify

The CLI supports four user roles: Administrator, Operator, Viewer, and Service. The privilege levels for each role are different; therefore, the menus and available commands vary for each role.

- **Administrators**—This user role has the highest level of privileges. Administrators have unrestricted view access and can perform the following functions:
  - Add users and assign passwords
  - Enable and disable control of physical interfaces and virtual sensors
  - Assign physical sensing interfaces to a virtual sensor
  - Modify the list of hosts allowed to connect to the sensor as a configuring or viewing agent
  - Modify sensor address configuration
  - Tune signatures
  - Assign configuration to a virtual sensor
  - Manage routers
- **Operators**—This user role has the second highest level of privileges. Operators have unrestricted view access and can perform the following functions:
  - Modify their passwords
  - Tune signatures
  - Manage routers
  - Assign configuration to a virtual sensor

- **Viewers**—This user role has the lowest level of privileges. Viewers can view configuration and event data and can modify their passwords.

Tip Monitoring applications only require viewer access to the sensor. You can use the CLI to set up a user account with viewer privileges and then configure the event viewer to use this account to connect to the sensor.

- **Service**—This user role does not have direct access to the CLI. Service account users are logged directly into a bash shell. Use this account for support and troubleshooting purposes only. Unauthorized modifications are not supported and will require the device to be reimaged to guarantee proper operation. You can create only one user with the service role.

  When you log in to the service account, you receive the following warning:

```
****************************** WARNING ****************************************
UNAUTHORIZED ACCESS TO THIS NETWORK DEVICE IS PROHIBITED.
This account is intended to be used for support and troubleshooting purposes only.
Unauthorized modifications are not supported and will require this device to be
re-imaged to guarantee proper operation.
*******************************************************************************
```

Note The service role is a special role that allows you to bypass the CLI if needed. Only a user with Administrator privileges can edit the service account.

CLI Behavior

Follow these tips when using the IPS CLI:

Prompts

- You cannot change the prompt displayed for the CLI commands.
- User interactive prompts occur when the system displays a question and waits for user input. The default input is displayed inside brackets [ ]. To accept the default input, press **Enter**.

Help

- To display the help for a command, type **?** after the command.

  The following example demonstrates the ? function:

```
sensor# configure ?
terminal     Configure from the terminal
sensor# configure
```

Note When the prompt returns from displaying help, the command previously entered is displayed without the ?.

- You can type **?** after an incomplete token to view the valid tokens that complete the command. If there is a trailing space between the token and the ?, you receive an ambiguous command error:

```
sensor# show c ?
% Ambiguous command : "show c"
```

If you enter the token without the space, a selection of available tokens for the completion (with no help description) appears:

```
sensor# show c?
clock configuration
sensor# show c
```

- Only commands available in the current mode are displayed by help.

Tab Completion

- Only commands available in the current mode are displayed by tab complete and help.
- If you are unsure of the complete syntax for a command, you can type a portion of the command and press **Tab** to complete the command.
- If multiple commands match for tab completion, nothing is displayed.

Recall

- To recall the commands entered in a mode, use the Up Arrow or Down Arrow keys or press **Ctrl-P** or **Ctrl-N**.

Note Help and tab complete requests are not reported in the recall list.

- A blank prompt indicates the end of the recall list.

Case Sensitivity

- The CLI is not case sensitive, but it does echo back the text in the same case you typed it. For example, if you type:

  ```
  sensor# CONF
  ```

  and press **Tab**, the sensor displays:

  ```
  sensor# CONFigure
  ```

Display Options

- `–More–` is an interactive prompt that indicates that the terminal output exceeds the allotted display space. To display the remaining output, press the **spacebar** to display the next page of output or press **Enter** to display the output one line at a time.
- To clear the current line contents and return to a blank command line, press **Ctrl-C**.

Command Line Editing

Table 1-1 describes the command line editing capabilities provided by the CLI.

Table 1-1 Command Line Editing

Keys	Description
Tab	Completes a partial command name entry. When you type a unique set of characters and press Tab, the system completes the command name. If you type a set of characters that could indicate more than one command, the system beeps to indicate an error. Type a question mark (?) immediately following the partial command (no space). The system provides a list of commands that begin with that string.
Backspace	Erases the character to the left of the cursor.
Enter	At the command line, pressing Enter processes a command. At the `---More---` prompt on a terminal screen, pressing Enter scrolls down a line.
Spacebar	Enables you to see more output on the terminal screen. Press the Spacebar when you see the line `---More---` on the screen to display the next screen.
Left arrow	Moves the cursor one character to the left. When you type a command that extends beyond a single line, you can press the Left Arrow key repeatedly to scroll back toward the system prompt and verify the beginning of the command entry.
Right arrow	Moves the cursor one character to the right.
Up Arrow or Ctrl-P	Recalls commands in the history buffer, beginning with the most recent command. Repeat the key sequence to recall successively older commands.
Down Arrow or Ctrl-N	Returns to more recent commands in the history buffer after recalling commands with the Up Arrow or Ctrl-P. Repeat the key sequence to recall successively more recent commands.
Ctrl-A	Moves the cursor to the beginning of the line.
Ctrl-B	Moves the cursor back one character.
Ctrl-D	Deletes the character at the cursor.
Ctrl-E	Moves the cursor to the end of the command line.
Ctrl-F	Moves the cursor forward one character.
Ctrl-K	Deletes all characters from the cursor to the end of the command line.
Ctrl-L	Clears the screen and redisplays the system prompt and command line
Ctrl-T	Transposes the character to the left of the cursor with the character located at the cursor.
Ctrl-U	Deletes all characters from the cursor to the beginning of the command line.
Ctrl-V	Inserts a code to indicate to the system that the keystroke immediately following should be treated as a command entry, *not* as an editing key.
Ctrl-W	Deletes the word to the left of the cursor.
Ctrl-Y	Recalls the most recent entry in the delete buffer. The delete buffer contains the last ten items you deleted or cut.
Ctrl-Z	Ends configuration mode and returns you to the EXEC prompt.
Esc-B	Moves the cursor back one word.
Esc-C	Capitalizes the word at the cursor.

Table 1-1 Command Line Editing (continued)

Keys	Description
Esc-D	Deletes from the cursor to the end of the word.
Esc-F	Moves the cursor forward one word.
Esc-L	Changes the word at the cursor to lowercase.
Esc-U	Capitalizes from the cursor to the end of the word.

IPS Command Modes

IPS CLI has the following command modes:

- privileged EXEC—Entered when you log in to the CLI interface.
- global configuration—Entered from privileged EXEC mode by typing **`configure terminal`**.

  The command prompt is `sensor(config)#`.
- service mode configuration—Entered from global configuration mode by typing **`service`** *service-name*.

  The command prompt is `sensor(config-ser)#`, where `ser` is the first three characters of the service name.
- multi-instance service mode—Entered from global configuration mode by typing **`service`** *service-name log-instance-name*.

  The command prompt is `sensor(config-log)#` where `log` is the first three characters of the log instance name. The only multi-instance services in the system are signature definition and event action rules.

Regular Expression Syntax

Regular expressions are text patterns that are used for string matching. Regular expressions contain a mix of plain text and special characters to indicate what kind of matching to do. For example, if you are looking for a numeric digit, the regular expression to search for is “[0-9]”. The brackets indicate that the character being compared should match any one of the characters enclosed within the bracket. The dash (-) between 0 and 9 indicates that it is a range from 0 to 9. Therefore, this regular expression will match any character from 0 to 9, that is, any digit.

To search for a specific special character, you must use a backslash before the special character. For example, the single character regular expression “*” matches a single asterisk.

The regular expressions defined in this section are similar to a subset of the POSIX Extended Regular Expression definitions. In particular, “[..]”, “[==]”, and “[::]” expressions are not supported. Also, escaped expressions representing single characters are supported. A character can be represented as its hexadecimal value, for example, \x61 equals ‘a,’ so \x61 is an escaped expression representing the character ‘a.’

Table 1-2 lists the special characters.

Table 1-2 Regular Expression Syntax

Character	Description
^	Beginning of the string. The expression "^A" will match an "A" only at the beginning of the string.
^	Immediately following the left-bracket ([). Excludes the remaining characters within brackets from matching the target string. The expression "[^0-9]" indicates that the target character should not be a digit.
$	Matches the end of the string. The expression "abc$" matches the sub-string "abc" only if it is at the end of the string.
|	Allows the expression on either side to match the target string. The expression "a|b" matches "a" as well as "b."
.	Matches any character.
*	Indicates that the character to the left of the asterisk in the expression should match 0 or more times.
+	Similar to * but there should be at least one match of the character to the left of the + sign in the expression.
?	Matches the character to its left 0 or 1 times.
()	Affects the order of pattern evaluation and also serves as a tagged expression that can be used when replacing the matched sub-string with another expression.
[]	Enclosing a set of characters indicates that any of the enclosed characters may match the target character.
\	Allows specifying a character that would otherwise be interpreted as special. \xHH represents the character whose value is the same as the value represented by (HH) hexadecimal digits [0-9A-Fa-f]. The value must be non-zero. BEL is the same as \x07, BS is \x08, FF is \x0C, LF is \x0A, CR is \x0D, TAB is \x09, and VT is \x0B. For any other character 'c', '\c' is the same as 'c' except that it is never interpreted as special

The following examples demonstrate the special characters:

- **`a*`** matches any number of occurrences of the letter a, including none.
- **`a+`** requires that at least one letter a be in the string to be matched.
- **`ba?b`** matches the string bb or bab.
- **`\**`** matches any number of asterisks (*).

To use multipliers with multiple-character patterns, you enclose the pattern in parentheses.

- **`(ab)*`** matches any number of the multiple-character string ab.
- **`([A-Za-z][0-9])+`** matches one or more instances of alphanumeric pairs, but not none (that is, an empty string is not a match).

The order for matches using multipliers (*, +, or ?) is to put the longest construct first. Nested constructs are matched from outside to inside. Concatenated constructs are matched beginning at the left side of the construct. Thus, the regular expression matches A9b3, but not 9Ab3 because the letters are specified before the numbers.

You can also use parentheses around a single- or multiple-character pattern to instruct the software to remember a pattern for use elsewhere in the regular expression.

To create a regular expression that recalls a previous pattern, you use parentheses to indicate memory of a specific pattern and a backslash (\) followed by a digit to reuse the remembered pattern. The digit specifies the occurrence of a parentheses in the regular expression pattern. If you have more than one remembered pattern in your regular expression, then \1 indicates the first remembered pattern, and \2 indicates the second remembered pattern, and so on.

The following regular expression uses parentheses for recall:

- **a(.)bc(.)\1\2** matches an *a* followed by any character, followed by *bc* followed by any character, followed by the first *any* character again, followed by the second *any* character again.

  For example, the regular expression can match aZbcTZT. The software remembers that the first character is Z and the second character is T and then uses Z and T again later in the regular expression.

General CLI Commands

The following CLI commands are generic to IPS 5.1.

- **configure terminal**—Enters global configuration mode.

  Global configuration commands apply to features that affect the system as a whole rather than just one protocol or interface.

  ```
  sensor# configure terminal
  sensor(config)#
  ```

- **service**—Takes you to the following configuration submodes: analysis-engine, authentication, event-action-rules, host, interface, logger, network-access, notification, signature-definition, ssh-known-hosts, trusted-certificates, and web-server.

  Note The event-action-rules and signature-definition submodes are multiple instance services. Only one predefined instance is allowed for each. For event-action-rules, the only supported instance name is rules0. For signature-definition, the only supported instance name is sig0.

  ```
  sensor# configure terminal
  sensor(config)# service event-action-rules rules0
  sensor(config-rul)#
  ```

- **end**—Exits configuration mode or any configuration submodes. It takes you back to the top-level EXEC menu.

  ```
  sensor# configure terminal
  sensor(config)# end
  sensor#
  ```

- **exit**—Exits any configuration mode or closes an active terminal session and terminates the EXEC mode. It takes you to the previous menu session.

```
sensor# configure terminal
sensor(config)# service event-action-rules rules0
sensor(config-rul)# exit
sensor(config)# exit
sensor#
```

CLI Keywords

In general, use the **no** form of a command to disable a feature or function. Use the command without the keyword **no** to enable a disabled feature or function. For example, the command **ssh host-key** *ipaddress* adds an entry to the known hosts table, the command **no ssh host-key** *ipaddress* removes the entry from the known hosts table. Refer to the individual commands for a complete description of what the **no** form of that command does.

Service configuration commands can also have a default form. Use the **default** form of the command to return the command setting to its default. This keyword applies to the **service** submenu commands used for application configuration. Typing **default** with the command resets the parameter to the default value. You can only use the **default** keyword with commands that specify a default value in the configuration files.

CHAPTER 2

Logging In to the Sensor

This chapter explains how to log in to the sensor. It contains the following sections:

Overview

The number of concurrent CLI sessions is limited based on the platform. IDS-4210, IDS-4215, and NM-CIDS are limited to three concurrent CLI session. All other platforms allow ten concurrent sessions.

Supported User Role

You can log in with the following user privileges:

- Administrator
- Operator
- Viewer
- Service

  The service role does not have direct access to the CLI. Service account users are logged directly into a bash shell. Use this account for support and troubleshooting purposes only. Unauthorized modifications are not supported and will require the sensor to be reimaged to guarantee proper operation. You can create only one user with the service role.

When you log in to the service account, you receive the following warning:

```
******************************** WARNING ****************************************
UNAUTHORIZED ACCESS TO THIS NETWORK DEVICE IS PROHIBITED.
This account is intended to be used for support and troubleshooting purposes only.
Unauthorized modifications are not supported and will require this device to be
re-imaged to guarantee proper operation.
*********************************************************************************
```

Note The service role is a special role that allows you to bypass the CLI if needed. Only a user with Administrator privileges can edit the service account.

Note For IPS 5.0 and later, you can no longer remove the **cisco** account. You can disable it using the **no password cisco** command, but you cannot remove it. To use the **no password cisco** command, there must be another administrator account on the sensor. Removing the **cisco** account through the service account is not supported. If you remove the **cisco** account through the service account, the sensor most likely will not boot up, so to recover the sensor you must reinstall the sensor system image.

Logging In to the Appliance

You can log in to the appliance from a console port or by connecting a monitor and keyboard to the sensor.

Note You must initialize the sensor (run the **setup** command) from the console. After networking is configured, SSH and Telnet are available. For the procedure, see Logging In to the Sensor, page 2-8.

To log in to the appliance, follow these steps:

Step 1 Log in to the appliance:

- Connect a console port to the sensor.

  For the procedure, see Connecting an Appliance to a Terminal Server, page 2-3.
- Connect a monitor and a keyboard to the sensor.

Step 2 Enter your username and password at the login prompt:

Note The default username and password are both **cisco**. You are prompted to change them the first time you log in to the appliance.You must first enter the UNIX password, which is **cisco**. Then you must enter the new password twice.

```
login: cisco
Password:
***NOTICE***
This product contains cryptographic features and is subject to United States
and local country laws governing import, export, transfer and use. Delivery
of Cisco cryptographic products does not imply third-party authority to import,
export, distribute or use encryption. Importers, exporters, distributors and
users are responsible for compliance with U.S. and local country laws. By using
this product you agree to comply with applicable laws and regulations. If you
are unable to comply with U.S. and local laws, return this product immediately.
```

```
A summary of U.S. laws governing Cisco cryptographic products may be found at:
http://www.cisco.com/wwl/export/crypto/tool/stqrg.html

If you require further assistance please contact us by sending email to
export@cisco.com.

***LICENSE NOTICE***
There is no license key installed on the system.
Please go to http://www.cisco.com/go/license
to obtain a new license or install a license.
ips-4240#
```

Connecting an Appliance to a Terminal Server

A terminal server is a router with multiple, low speed, asynchronous ports that are connected to other serial devices. You can use terminal servers to remotely manage network equipment, including appliances.

To set up a Cisco terminal server with RJ-45 or hydra cable assembly connections, follow these steps:

Step 1 Connect to a terminal server using one of the following methods:

- For IDS-4215, IPS-4240, IPS-4255, and IPS-4260:
  - For RJ-45 connections, connect a 180/rollover cable from the console port on the appliance to a port on the terminal server.
  - For hydra cable assemblies, connect a straight-through patch cable from the console port on the appliance to a port on the terminal server.
- For all other appliances, connect the M.A.S.H. adapter (part number 29-4077-01) to COM1 on the appliance and:
  - For RJ-45 connections, connect a 180/rollover cable from the M.A.S.H. adapter to a port on the terminal server.
  - For hydra cable assemblies, connect a straight-through patch cable from the M.A.S.H. adapter to a port on the terminal server.

Step 2 Configure the line/port on the terminal server as follows:

a. In enable mode, enter the following configuration, where # is the line number of the port to be configured:

```
config t
line #
login
transport input all
stopbits 1
flowcontrol hardware
speed 9600
exit
exit
wr mem
```

b. If you are configuring a terminal server for an IDS-4215, IPS-4240, IPS-4255, or IPS-4260, skip to Step 3.

Otherwise, for all other supported appliances, to direct all output to the terminal server, log in to the CLI and enter the following commands:

```
sensor# configure terminal
sensor(config)# display-serial
```

Output is directed to the serial port. Use the **no display-serial** command to redirect output to the keyboard/monitor.

Note You can set up a terminal server and use the **display-serial** command to direct all output from the appliance to the serial port. This option lets you view system messages on a console connected to the serial port, even during the boot process. When you use this option, all output is directed to the serial port and any local keyboard/monitor connection is disabled. However, BIOS and POST messages are still displayed on the local keyboard/monitor. For the procedure, see Directing Output to a Serial Connection, page 13-21.

Note There is only one console port on an IDS-4215, IPS-4240, IPS-4255, and IPS-4260; therefore, the **display-serial** and **no display-serial** commands do not apply to those platforms.

Step 3 Be sure to properly close a terminal session to avoid unauthorized access to the appliance.

If a terminal session is not stopped properly, that is, if it does not receive an exit(0) signal from the application that initiated the session, the terminal session can remain open. When terminal sessions are not stopped properly, authentication is not performed on the next session that is opened on the serial port.

Caution Always exit your session and return to a login prompt before terminating the application used to establish the connection.

Caution If a connection is dropped or terminated by accident, you should reestablish the connection and exit normally to prevent unauthorized access to the appliance.

Logging In to IDSM-2

You can log in to IDSM-2 from the switch.

Note You must initialize the sensor (run the **setup** command) from the switch. After networking is configured, SSH and Telnet are available.

To session in to IDSM-2, follow these steps

Step 1 Session to IDSM-2 from the switch:

- For Catalyst Software:

  console>(enable) **session** *slot_number*

- For Cisco IOS software:

  router# **session** *slot_number* **processor 1**

Step 2 Enter your username and password at the login prompt:

Note The default username and password are both **cisco**. You are prompted to change them the first time you log in to IDSM-2.You must first enter the UNIX password, which is **cisco**. Then you must enter the new password twice.

```
login: cisco
Password:
***NOTICE***
This product contains cryptographic features and is subject to United States
and local country laws governing import, export, transfer and use. Delivery
of Cisco cryptographic products does not imply third-party authority to import,
export, distribute or use encryption. Importers, exporters, distributors and
users are responsible for compliance with U.S. and local country laws. By using
this product you agree to comply with applicable laws and regulations. If you
are unable to comply with U.S. and local laws, return this product immediately.

A summary of U.S. laws governing Cisco cryptographic products may be found at:
http://www.cisco.com/wwl/export/crypto/tool/stqrg.html

If you require further assistance please contact us by sending email to
export@cisco.com.

***LICENSE NOTICE***
There is no license key installed on the system.
Please go to http://www.cisco.com/go/license
to obtain a new license or install a license.
idsm-2#
```

Logging In to NM-CIDS

You cannot access NM-CIDS through the router using the **session** command until you assign the internal management port an IP address. You can do this either by assigning an IP address directly to the IDS interface or by assigning an unnumbered loopback interface. If you do not set an IP address before you session to NM-CIDS, you receive an error message.

After you assign the IP address, you can log in to NM-CIDS from the router console.

Note You must initialize the sensor (run the **setup** command) from the router. After networking is configured, SSH and Telnet are available.

To session to NM-CIDS, follow these steps

Step 1 Assign an IP address to the management port:

a. Assign an IP address directly to the IDS interface:

```
router(config)# interface IDS-Sensor1/0
router(config-if)# ip unnumbered Loopback0
router(config-if)# exit
```

b. Assign an unnumbered loopback interface:

```
router(config)# interface Loopback0
router(config-if)# ip address 1.2.3.4 255.255.255.255
router(config-if)# exit
```

The IP address can be any address that is not used anywhere else in the network.

Step 2 Session to NM-CIDS through the router console:

service-module IDS-Sensor *slot_number*/**0 session**

Step 3 Enter your username and password at the login prompt:

Note The default username and password are both **cisco**. You are prompted to change them the first time you log in to NM-CIDS. You must first enter the UNIX password, which is **cisco**. Then you must enter the new password twice.

```
login: cisco
Password:
***NOTICE***
This product contains cryptographic features and is subject to United States
and local country laws governing import, export, transfer and use. Delivery
of Cisco cryptographic products does not imply third-party authority to import,
export, distribute or use encryption. Importers, exporters, distributors and
users are responsible for compliance with U.S. and local country laws. By using
this product you agree to comply with applicable laws and regulations. If you
are unable to comply with U.S. and local laws, return this product immediately.

A summary of U.S. laws governing Cisco cryptographic products may be found at:
http://www.cisco.com/wwl/export/crypto/tool/stqrg.html

If you require further assistance please contact us by sending email to
export@cisco.com.

***LICENSE NOTICE***
There is no license key installed on the system.
Please go to http://www.cisco.com/go/license
to obtain a new license or install a license.
nm-cids#
```

Step 4 Press **Ctrl-Shift-6**, then **x** to get back to the router prompt if you sessioned to the router.

Logging In to AIP-SSM

You can log in to AIP-SSM from ASA.

Note You must initialize the sensor (run the **setup** command) from ASA. After networking is configured, SSH and Telnet are available.

To log in to AIP-SSM from ASA, follow these steps:

Step 1 Log in to ASA.

Note If ASA is operating in multi-mode, use the **change system** command to get to the system level prompt before continuing.

Step 2 Session to AIP-SSM:

```
asa# session 1
Opening command session with slot 1.
Connected to slot 1. Escape character sequence is 'CTRL-^X'.
```

You have 60 seconds to log in before the session times out.

Step 3 Enter your username and password at the login prompt:

Note The default username and password are both **cisco**. You are prompted to change them the first time you log in to AIP-SSM. You must first enter the UNIX password, which is **cisco**. Then you must enter the new password twice.

```
login: cisco
Password:
***NOTICE***
This product contains cryptographic features and is subject to United States
and local country laws governing import, export, transfer and use. Delivery
of Cisco cryptographic products does not imply third-party authority to import,
export, distribute or use encryption. Importers, exporters, distributors and
users are responsible for compliance with U.S. and local country laws. By using
this product you agree to comply with applicable laws and regulations. If you
are unable to comply with U.S. and local laws, return this product immediately.

A summary of U.S. laws governing Cisco cryptographic products may be found at:
http://www.cisco.com/wwl/export/crypto/tool/stqrg.html

If you require further assistance please contact us by sending email to
export@cisco.com.

***LICENSE NOTICE***
There is no license key installed on the system.
Please go to http://www.cisco.com/go/license
to obtain a new license or install a license.
aip-ssm#
```

Step 4 To escape from a session and return to the ASA prompt, do one of the following:

- Enter **exit**.
- Enter the **CTRL-Shift-6-x** sequence (represented as **CTRL^X**).

Logging In to the Sensor

After you have initialized the sensor using the **setup** command and enabled Telnet, you can use SSH or Telnet to log in to the sensor.

To log in to the sensor, follow these steps:

Step 1 To log in to the sensor over the network using SSH or Telnet:

```
ssh sensor_ip_address
telnet sensor_ip_address
```

Step 2 Enter your username and password at the login prompt:

```
login: ******
Password: ******
***NOTICE***
This product contains cryptographic features and is subject to United States
and local country laws governing import, export, transfer and use. Delivery
of Cisco cryptographic products does not imply third-party authority to import,
export, distribute or use encryption. Importers, exporters, distributors and
users are responsible for compliance with U.S. and local country laws. By using
this product you agree to comply with applicable laws and regulations. If you
are unable to comply with U.S. and local laws, return this product immediately.

A summary of U.S. laws governing Cisco cryptographic products may be found at:
http://www.cisco.com/wwl/export/crypto/tool/stqrg.html

If you require further assistance please contact us by sending email to
export@cisco.com.

***LICENSE NOTICE***
There is no license key installed on the system.
Please go to http://www.cisco.com/go/license
to obtain a new license or install a license.
sensor#
```

CHAPTER 3

Initializing the Sensor

This chapter explains how to initialize the sensor using the **setup** command. It contains the following sections:

Overview

After you have installed the sensor on your network, you must use the **setup** command to initialize it. With the **setup** command, you configure basic sensor settings, including the hostname, IP interfaces, Telnet server, web server port, access control lists, time settings, and assign and enable interfaces. After you have initialized the sensor, you can communicate with it over the network. You are then ready to configure intrusion prevention.

System Configuration Dialog

When you enter the **setup** command, an interactive dialog called the System Configuration Dialog appears on the system console screen. The System Configuration Dialog guides you through the configuration process.

The values shown in brackets next to each prompt are the current values.

You must go through the entire System Configuration Dialog until you come to the option that you want to change. To accept default settings for items that you do not want to change, press **Enter**.

To return to the EXEC prompt without making changes and without going through the entire System Configuration Dialog, press **Ctrl-C**.

The System Configuration Dialog also provides help text for each prompt. To access the help text, press the question mark (?) key at a prompt.

When you complete your changes, the System Configuration Dialog shows you the configuration that you created during the setup session. It also asks you if you want to use this configuration. If you enter **`yes`**, the configuration is saved. If you enter **`no`**, the configuration is not saved and the process begins again. There is no default for this prompt; you must enter either **`yes`** or **`no`**.

You can configure daylight savings time either in recurring mode or date mode. If you select recurring mode, the start and end days are based on week, day, month, and time. If you select date mode, the start and end days are based on month, day, year, and time. Selecting Disable turns off daylight savings time.

You can edit the default virtual sensor, vs0, through the System Configuration Dialog. You can assign promiscuous and/or inline-pairs to the virtual sensor. This also enables the assigned interfaces. After setup is complete, the virtual sensor is configured to monitor traffic.

Note You only need to set the date and time in the System Configuration Dialog if the system is an appliance and is NOT using NTP.

Initializing the Sensor

To initialize the sensor, follow these steps:

Step 1 Log in to the sensor using an account with administrator privileges:

- Log in to the appliance by using a serial connection or with a monitor and keyboard.

  Note You cannot use a monitor and keyboard with IDS-4215, IPS-4240, IPS-4255, or IPS-4260.

- Session to IDSM-2:
  - For Catalyst software:

    ```
    console> enable
    console> (enable) session module_number
    ```

  - For Cisco IOS software:

    ```
    Router# session slot slot_number processor 1
    ```

- Session to NM-CIDS:

  ```
  router# service-module IDS-Sensor slot_number/port_number session
  ```

- Session to AIP-SSM:

  ```
  asa# session 1
  ```

Note The default username and password are both **cisco**.

Step 2 The first time you log in to the sensor you are prompted to change the default password.

Passwords must be at least eight characters long and be strong, that is, not be a dictionary word.

Caution If you forget your password, you may have to reimage your sensor unless there is another user with Administrator privileges (see Chapter 17, "Upgrading, Downgrading, and Installing System Images"). The other Administrator can log in and assign a new password to the user who forgot the password. Or, if you have created the service account for support purposes, you can have TAC create a password. For more information, see Creating the Service Account, page 4-13.

After you change the password, the `sensor#` prompt appears.

Step 3 Enter the **`setup`** command.

The System Configuration Dialog is displayed.

Note The System Configuration Dialog is an interactive dialog. The default settings are displayed.

```
    --- System Configuration Dialog ---


At any point you may enter a question mark '?' for help.
User ctrl-c to abort configuration dialog at any prompt.
Default settings are in square brackets '[]'.


Current Configuration:


service host
network-settings
host-ip 10.1.9.201/24,10.1.9.1
host-name sensor
telnet-option disabled
ftp-timeout 300
login-banner-text
exit
time-zone-settings
offset 0
standard-time-zone-name UTC
exit
summertime-option disabled
ntp-option disabled
exit
service web-server
port 443
exit


Current time: Wed May 5 10:25:35 2004
```

Step 4 Press the spacebar to get to the following question:

```
Continue with configuration dialog?[yes]:
```

Press the spacebar to show one page at a time. Press **Enter** to show one line at a time.

Step 5 Enter **`yes`** to continue.

Step 6 Specify the hostname.

The hostname is a case-sensitive character string up to 64 characters. Numbers, "_" and "-" are valid, but spaces are not acceptable. The default is sensor.

Step 7 Specify the IP interface.

The IP interface is in the form of IP Address/Netmask,Gateway: *X.X.X.X/nn*,*Y.Y.Y.Y*, where *X.X.X.X* specifies the sensor IP address as a 32-bit address written as 4 octets separated by periods, *nn* specifies the number of bits in the netmask, and *Y.Y.Y.Y* specifies the default gateway as a 32-bit address written as 4 octets separated by periods.

Step 8 Specify the Telnet server status.

You can disable or enable Telnet services. The default is disabled.

Step 9 Specify the web server port.

The web server port is the TCP port used by the web server (1 to 65535). The default is 443.

Note If you change the web server port, you must specify the port in the URL address of your browser when you connect to the IDM in the format `https://`*sensor_ip_ address*`:port` (for example, `https://10.1.9.201:1040`).

Note The web server is configured to use TLS/SSL encryption by default. Setting the port to 80 does not disable the encryption.

Step 10 Enter **yes** to modify the network access list.

a. If you want to delete an entry, enter the number of the entry and press **Enter**, or press **Enter** to get to the Permit line.

b. Enter the IP address and netmask of the network you want to add to the access list.

The IP network interface is in the form of IP Address/Netmask: *X.X.X.X/nn*, where *X.X.X.X* specifies the network IP address as a 32-bit address written as 4 octets separated by periods and *nn* specifies the number of bits in the netmask for that network.

For example, 10.0.0.0/8 permits all IP addresses on the 10.0.0.0 network (10.0.0.0-10.255.255.255) and 10.1.1.0/24 permits only the IP addresses on the 10.1.1.0 subnet (10.1.1.0-10.1.1.255).

If you want to permit access to a single IP address than the entire network, use a 32-bit netmask. For example, 10.1.1.1/32 permits just the 10.1.1.1 address.

c. Repeat Step b until you have added all networks that you want to add to the access list.

d. Press **Enter** at a blank permit line to proceed to the next step.

Step 11 Enter **yes** to modify the system clock settings.

a. Enter **yes** if you want to use NTP.

You will need the NTP server IP address, the NTP key ID, and the NTP key value. If you do not have those at this time, you can configure NTP later. For the procedure, see Configuring the Sensor to Use an NTP Time Source, page 4-30.

b. Enter **yes** to modify summertime settings.

Note Summertime is also known as DST. If your location does not use Summertime, go to Step n.

c. Choose recurring, date, or disable to specify how you want to configure summertime settings.

The default is recurring.

d. If you chose recurring, specify the month you want to start summertime settings.

Valid entries are january, february, march, april, may, june, july, august, september, october, november, and december.

The default is april.

e. Specify the week you want to start summertime settings.

   Valid entries are first, second, third, fourth, fifth, and last.

   The default is first.

f. Specify the day you want to start summertime settings.

   Valid entries are sunday, monday, tuesday, wednesday, thursday, friday, and saturday.

   The default is sunday.

g. Specify the time you want to start summertime settings.

   The default is 02:00:00.

Note The default recurring summertime parameters are correct for time zones in the United States. The default values specify a start time of 2 a.m. on the first Sunday in April, and a stop time of 2 a.m. on the fourth Sunday in October. The default summertime offset is 60 minutes.

h. Specify the month you want summertime settings to end.

   Valid entries are january, february, march, april, may, june, july, august, september, october, november, and december.

   The default is october.

i. Specify the week you want the summertime settings to end.

   Valid entries are first, second, third, fourth, fifth, and last.

   The default is last.

j. Specify the day you want the summertime settings to end.

   Valid entries are sunday, monday, tuesday, wednesday, thursday, friday, and saturday.

   The default is sunday.

k. Specify the time you want summertime settings to end.

l. Specify the DST zone.

   The zone name is a character string up to 24 characters long in the pattern [A-Za-z0-9()+:,_/-]+$.

m. Specify the summertime offset.

   Specify the summertime offset from UTC in minutes (negative numbers represent time zones west of the Prime Meridian).

   The default is 0.

n. Enter `yes` to modify the system time zone.

o. Specify the standard time zone name.

   The zone name is a character string up to 24 characters long.

p. Specify the standard time offset.

   The default is 0.

   Specify the standard time zone offset from UTC in minutes (negative numbers represent time zones west of the Prime Meridian).

Step 12 Enter `yes` to modify the virtual sensor configuration (vs0).

The current interface configuration appears:

```
Current interface configuration
  Command control: GigabitEthernet0/1
  Unused:
    GigabitEthernet0/0
    GigabitEthernet2/1
    GigabitEthernet2/0
  Promiscuous:
    None
  Inline:
    None
  Inline VLAN Pair:
    None
```

Step 13 Enter **yes** to add a promiscuous or monitoring interface.

Step 14 Enter the interface you want to add, for example, **GigabitEthernet0/1**.

Step 15 Enter **yes** to add inline interface pairs (appears only if your platform supports inline interface pairs).

a. Enter the inline interface pair name.

b. Enter the inline interface pair description.

The default is `Created via setup by user <yourusername>`.

c. Enter the name of the first interface in the inline pair, **interface1**.

d. Enter the name of the second interface in the inline pair, **interface2**.

e. Repeat Steps a through d to add another inline interface pair, or press **Enter** for the next option.

Step 16 Enter **yes** to add inline VLAN pairs (appears only if your platform supports inline VLAN pairs).

A list of interfaces available for inline VLAN pairs appears:

```
Available Interfaces:
 [1] GigabitEthernet0/0
 [2] GigabitEthernet2/0
 [3] GigabitEthernet2/1
```

Step 17 Enter the number of the interface you want to subdivide into inline VLAN pairs.

The current inline VLAN pair configuration for that interface appears:

```
Inline Vlan Pairs for GigabitEthernet0/0
  None
```

a. Enter the subinterface number to add.

b. Enter the inline VLAN pair description.

c. Enter the first VLAN number (vlan1).

d. Enter the second VLAN number (vlan2).

e. Repeat Steps a through d to add another inline VLAN pair on this interface or press **Enter** for the next option.

Step 18 Enter **yes** to subdivide another interface. Enter **no** or press **Enter** to complete the addition of the inline VLAN pairs.

Your configuration appears with the following options:

```
[0] Go to the command prompt without saving this config.
[1] Return back to the setup without saving this config.
[2] Save this configuration and exit setup.
```

Step 19 Enter **2** to save the configuration.

```
Enter your selection[2]: 2
Configuration Saved.
```

Step 20 Enter **yes** to modify the system date and time.

Note This option is not available on modules or when NTP has been configured. The modules get their time from the router or switch in which they are installed, or from the configured NTP server.

a. Enter the local date (yyyy-mm-dd).

b. Enter the local time (hh:mm:ss).

Step 21 Reboot the sensor:

```
sensor# reset
Warning: Executing this command will stop all applications and reboot the node.
Continue with reset? []:
```

Step 22 Enter **yes** to continue the reboot.

Step 23 Display the self-signed X.509 certificate (needed by TLS):

```
sensor# show tls fingerprint
MD5: C4:BC:F2:92:C2:E2:4D:EB:92:0F:E4:86:53:6A:C6:01
SHA1: 64:9B:AC:DE:21:62:0C:D3:57:2E:9B:E5:3D:04:8F:A7:FD:CD:6F:27
```

Step 24 Write down the certificate fingerprints.

You will need these to check the authenticity of the certificate when connecting to this sensor with a web browser.

Step 25 Apply the most recent service pack and signature update.

For information on how to obtain the most recent software, see Obtaining Cisco IPS Software, page 18-1. The Readme explains how to apply the most recent software update.

You are now ready to configure your sensor for intrusion prevention.

Verifying Initialization

After you have run the **setup** command, you should verify that your sensor has been initialized correctly.

To verify that you initialized your sensor, follow these steps:

Step 1 Log in to the sensor.

For the procedure, see Chapter 2, "Logging In to the Sensor."

Step 2 View your configuration:

```
sensor# show configuration
generating current config:
! ------------------------------
! Version 5.1(1)
! Current configuration last modified Wed Jun 29 19:18:14 2005
! ------------------------------
display-serial
```

```
! ------------------------------
service interface
physical-interfaces GigabitEthernet0/0
admin-state enabled
exit
physical-interfaces GigabitEthernet2/0
admin-state enabled
exit
physical-interfaces GigabitEthernet2/1
admin-state enabled
exit
bypass-mode auto
interface-notifications
missed-percentage-threshold 19
notification-interval 36
idle-interface-delay 33
exit
exit
! ------------------------------
service analysis-engine
virtual-sensor vs0
physical-interface GigabitEthernet0/0 subinterface-number 0
physical-interface GigabitEthernet2/1
exit
exit
! ------------------------------
service authentication
exit
! ------------------------------
service event-action-rules rules0
exit
! ------------------------------
service host
network-settings
host-ip 10.89.149.27/25,10.89.149.126
host-name sensor
telnet-option enabled
access-list 10.0.0.0/8
access-list 64.0.0.0/8
access-list 171.0.0.0/8
exit
time-zone-settings
offset 0
standard-time-zone-name UTC
exit
exit
! ------------------------------
service logger
exit
! ------------------------------
service network-access
exit
! ------------------------------
service notification
exit
! ------------------------------
service signature-definition sig0
signatures 2000 0
alert-severity high
status
enabled true
exit
alert-frequency
summary-mode fire-all
```

```
exit
exit
exit
signatures 2004 0
alert-severity low
status
enabled true
exit
alert-frequency
summary-mode fire-all
exit
exit
exit
signatures 3201 1
alert-frequency
summary-mode fire-all
exit
exit
exit
signatures 3301 0
status
enabled true
exit
exit
signatures 3401 0
status
enabled true
retired false
exit
engine string-tcp
event-action produce-alert|request-block-host
exit
alert-frequency
summary-mode fire-all
exit
exit
exit
exit
! ------------------------------
service ssh-known-hosts
exit
! ------------------------------
service trusted-certificates
trusted-certificates 10.89.149.227:443 certificate MIICJDCCAY0CCPy71vhtAwyNMA0GC
SqGSIb3DQEBBQUAMFcxCzAJBgNVBAYTAlVTMRwwGgYDVQQKExNDaXNjbyBTeXN0ZW1zLCBJbmMuMRIwE
AYDVQQLEwlTU00tSVBTMTAxFjAUBgNVBAMTDTEwLjg5LjE0OS4yMjcwHhcNMDUwNjE0MDUwODA3WhcNM
DcwNjE1MDUwODA3WjBXMQswCQYDVQQGEwJVUzEcMBoGA1UEChMTQ2lzY28gU3lzdGVtcywgSW5jLjESM
BAGA1UECxMJU1NNLUlQUzEwMRYwFAYDVQQDEw0xMC44OS4xNDkuMjI3MIGfMA0GCSqGSIb3DQEBAQUAA
4GNADCBiQKBgQCoOobDuZOEPudw63Rlt8K1YsymzR/D9Rlcnad/U0gjAQGfcUh3sG3TXPQewon1fH0+A
nBw8Jxv/ovSB1HJ3ujh5k7BrrB2QMv73ESsBDdxLY6SoX/yYANMf4zPcPCAORJ6DMQHFj44A+3tMZWsC
yaod23S1oYOxx7v5puPDYn3IQIDAQABMA0GCSqGSIb3DQEBBQUAA4GBAHfPM7jawvdfXkYyazqvy3ZOK
kHVwHji12vBLo+biULJG95hbTFlqO+ba3R6nPD3tepgx5zTdOr2onnlFHWD95Ii+PKdUxj7vfDBG8atn
obsEBJ1lAQDiogskdCs4ax1tB4SbEU5y1tktKgcwWEdJpbbNJhzpoRsRICfM3HlOEwN
exit
! ------------------------------
service web-server
exit
sensor#
```

Note You can also use the **more current-config** command to view your configuration.

Step 3 Display the self-signed X.509 certificate (needed by TLS):

```
sensor# show tls fingerprint
MD5: C4:BC:F2:92:C2:E2:4D:EB:92:0F:E4:86:53:6A:C6:01
SHA1: 64:9B:AC:DE:21:62:0C:D3:57:2E:9B:E5:3D:04:8F:A7:FD:CD:6F:27
```

Step 4 Write down the certificate fingerprints.

You will need these to check the authenticity of the certificate when connecting to this sensor with a web browser.

CHAPTER 4

Initial Configuration Tasks

This chapter contains procedures for the initial configuration tasks, such as changing sensor setup information, adding and deleting users, configuring time and setting up NTP, creating a service account, configuring SSH and TLS, and installing the license key.

Note You configured most of these settings when you initialized the sensor using the **setup** command.

This chapter contains the following sections:

Changing Network Settings

After you initialize your sensor, you may need to change some of the network settings that you configured when you ran the **setup** command. This section contains the following topics:

Changing the Hostname

Use the **host-name** *host_name* command in the service host submode to change the hostname of the sensor after you have run the **setup** command. The default is `sensor`.

Note The CLI prompt of the current session and other existing sessions will not be updated with the new hostname. Subsequent CLI login sessions will reflect the new hostname in the prompt.

To change the sensor hostname, follow these steps:

Step 1 Log in to the sensor using an account with administrator privileges.

Step 2 Enter network settings submode:

```
sensor# configure terminal
sensor(config)# service host
sensor(config-hos)# network-settings
```

Step 3 Change the sensor hostname:

```
sensor(config-hos-net)# host-name firesafe
```

Step 4 Verify the new hostname:

```
sensor(config-hos-net)# show settings
   network-settings
   ----------------------------------------------
      host-ip: 10.89.130.108/23,10.89.130.1 default:
      10.1.9.201/24,10.1.9.1
      host-name: firesafe default: sensor
      telnet-option: enabled default: disabled
      access-list (min: 0, max: 512, current: 1)
      ----------------------------------------------
         network-address: 0.0.0.0/0
         ----------------------------------------------
      ----------------------------------------------
      ftp-timeout: 300 seconds <defaulted>
      login-banner-text: <defaulted>
   ----------------------------------------------
sensor(config-hos-net)#
```

Step 5 To change the hostname back to the default setting, use the **default** form of the command:

```
sensor(config-hos-net)# default host-name
```

Step 6 Verify the change to the default hostname sensor:

```
sensor(config-hos-net)# show settings
   network-settings
   ----------------------------------------------
      host-ip: 10.89.130.108/23,10.89.130.1 default:
      10.1.9.201/24,10.1.9.1
      host-name: sensor <defaulted>
      telnet-option: enabled default: disabled
      access-list (min: 0, max: 512, current: 1)
      ----------------------------------------------
         network-address: 0.0.0.0/0
         ----------------------------------------------
      ----------------------------------------------
```

```
      ftp-timeout: 300 seconds <defaulted>
      login-banner-text: <defaulted>
   ---------------------------------------------
sensor(config-hos-net)#
```

Step 7 Exit network settings mode:

```
sensor(config-hos-net)# exit
sensor(config-hos)# exit
Apply Changes:?[yes]:
```

Step 8 Press **Enter** to apply the changes or enter **no** to discard them.

Changing the IP Address, Netmask, and Gateway

Use the **host-ip** *ip_address/netmask,default_gateway* command in the service host submode to change the IP address, netmask, and default gateway after you have run the **setup** command. The default is 10.1.9.201/24,10.1.9.1.

The **host-ip** is in the form of IP Address/Netmask/Gateway: X.X.X.X/nn,Y.Y.Y.Y, where X.X.X.X specifies the sensor IP address as a 32-bit address written as 4 octets separated by periods where X = 0-255, nn specifies the number of bits in the netmask, and Y.Y.Y.Y specifies the default gateway as a 32-bit address written as 4 octets separated by periods where Y = 0-255.

To change the sensor IP address, netmask, and default gateway, follow these steps:

Step 1 Log in to the sensor using an account with administrator privileges.

Step 2 Enter network settings mode:

```
sensor# configure terminal
sensor(config)# service host
sensor(config-hos)# network-settings
```

Step 3 Change the sensor IP address, netmask, and default gateway:

```
sensor(config-hos-net)# host-ip 10.89.146.110/24,10.89.146.254
```

Note The default gateway must be in the same subnet as the sensor's IP address or the sensor will generate an error and not accept the configuration change.

Step 4 Verify the new information:

```
sensor(config-hos-net)# show settings
   network-settings
   ---------------------------------------------
      host-ip: 10.89.146.110/24,10.89.146.254
      default: 10.1.9.201/24,10.1.9.1
      host-name: sensor default: sensor
      telnet-option: enabled default: disabled
      access-list (min: 0, max: 512, current: 1)
      ---------------------------------------------
         network-address: 0.0.0.0/0
         ---------------------------------------------
      ---------------------------------------------
```

```
      ftp-timeout: 300 seconds <defaulted>
      login-banner-text: <defaulted>
   ---------------------------------------------
```

Step 5 To change the information back to the default setting, use the **default** form of the command:

```
sensor(config-hos-net)# default host-ip
```

Step 6 Verify that the host IP is now the default of 10.1.9.201/24,10.1.9.1:

```
sensor(config-hos-net)# show settings
   network-settings
   ---------------------------------------------
      host-ip: 10.1.9.201/24,10.1.9.1 <defaulted>
      host-name: sensor default: sensor
      telnet-option: enabled default: disabled
      access-list (min: 0, max: 512, current: 1)
      ---------------------------------------------
         network-address: 0.0.0.0/0
         ---------------------------------------------
      ---------------------------------------------
      ftp-timeout: 300 seconds <defaulted>
      login-banner-text: <defaulted>
   ---------------------------------------------
sensor(config-hos-net)#
```

Step 7 Exit network settings mode:

```
sensor(config-hos-net)# exit
sensor(config-hos)# exit
Apply Changes:?[yes]:
```

Step 8 Press **Enter** to apply the changes or enter **no** to discard them.

Enabling and Disabling Telnet

Use the **telnet-option {enabled | disabled}** command in the service host submode to enable Telnet for remote access to the sensor. The default is disabled.

Caution Telnet is not a secure access service and therefore is disabled by default. However, SSH is always running on the sensor and it is a secure service.

To enable or disable Telnet services, follow these steps:

Step 1 Log in to the sensor using an account with administrator privileges.

Step 2 Enter network settings mode:

```
sensor# configure terminal
sensor(config)# service host
sensor(config-hos)# network-settings
```

Step 3 Enable Telnet services:

```
sensor(config-hos-net)# telnet-option enabled
sensor(config-hos-net)#
```

Step 4 Verify that Telnet is enabled:

```
sensor(config-hos-net)# show settings
  network-settings
  ----------------------------------------------
    host-ip: 10.89.130.108/23,10.89.130.1
    default: 10.1.9.201/24,10.1.9.1
    host-name: sensor default: sensor
    telnet-option: enabled default: disabled
    access-list (min: 0, max: 512, current: 1)
    ---------------------------------------------
       network-address: 0.0.0.0/0
       ---------------------------------------------
    ---------------------------------------------
    ftp-timeout: 300 seconds <defaulted>
    login-banner-text: <defaulted>
----------------------------------------------
sensor(config-hos-net)#
```

Step 5 Exit network settings mode:

```
sensor(config-hos-net)# exit
sensor(config-hos)# exit
Apply Changes:?[yes]:
```

Step 6 Press **Enter** to apply the changes or enter **no** to discard them.

Note To Telnet to the sensor, you must enable Telnet and configure the access list to allow the Telnet clients to connect. For the procedure, see Changing the Access List, page 4-5.

Changing the Access List

Use the **access-list** *ip_address/netmask* command in the service host submode to configure the access list, the list of hosts or networks that you want to have access to your sensor. Use the **no** form of the command to remove an entry from the list. The default access list is empty.

The following hosts must have an entry in the access list:

- Hosts that need to Telnet to your sensor.
- Hosts that need to use SSH with your sensor.
- Hosts, such as IDM, that need to access your sensor from a web browser.
- Management stations, such as VMS, that need access to your sensor.
- If your sensor is a master blocking sensor, the IP addresses of the blocking forwarding sensors must have an entry in the list.

To modify the access list, follow these steps:

Step 1 Log in to the sensor using an account with administrator privileges.

Step 2 Enter network settings mode:

```
sensor# configure terminal
sensor(config)# service host
sensor(config-hos)# network-settings
```

Step 3 Add an entry to the access list:

```
sensor(config-hos-net)# access-list 10.89.146.110/32
```

The netmask for a single host is 32.

Step 4 Verify the change you made to the access-list:

```
sensor(config-hos-net)# show settings
   network-settings
   ----------------------------------------------
      host-ip: 10.1.9.201/24,10.1.9.1 <defaulted>
      host-name: sensor <defaulted>
      telnet-option: enabled default: disabled
      access-list (min: 0, max: 512, current: 2)
      ---------------------------------------------
         network-address: 10.1.9.0/24
         ---------------------------------------------
         network-address: 10.89.146.110/32
         ---------------------------------------------
      ---------------------------------------------
      ftp-timeout: 300 seconds <defaulted>
      login-banner-text: <defaulted>
   ----------------------------------------------
```

Step 5 Remove the entry from the access list:

```
sensor(config-hos-net)# no access-list 10.89.146.110/32
```

Step 6 Verify the entry has been removed:

```
sensor(config-hos-net)# show settings
   network-settings
   ----------------------------------------------
      host-ip: 10.1.9.201/24,10.1.9.1 <defaulted>
      host-name: sensor <defaulted>
      telnet-option: enabled default: disabled
      access-list (min: 0, max: 512, current: 1)
      ---------------------------------------------
         network-address: 10.1.9.0/24
         ---------------------------------------------
      ---------------------------------------------
      ftp-timeout: 300 seconds <defaulted>
      login-banner-text: <defaulted>
   ----------------------------------------------
sensor(config-hos-net)#
```

The host is no longer in the list.

Step 7 Change the value back to the default:

```
sensor(config-hos-net)# default access-list
```

Step 8 Verify the value has been set back to the default:

```
sensor(config-hos-net)# show settings
   network-settings
   ----------------------------------------------
      host-ip: 10.89.130.108/23,10.89.130.1
      default: 10.1.9.201/24,10.1.9.1
      host-name: sensor <defaulted>
      telnet-option: enabled default: disabled
      access-list (min: 0, max: 512, current: 0)
      ---------------------------------------------
      ---------------------------------------------
```

```
      ftp-timeout: 300 seconds <defaulted>
      login-banner-text: <defaulted>
   ---------------------------------------------
sensor(config-hos-net)#
```

There are no hosts or networks in the list.

Step 9 Exit network settings mode:

```
sensor(config-hos-net)# exit
sensor(config-hos)# exit
Apply Changes:?[yes]:
```

Step 10 Press **Enter** to apply the changes or enter **no** to discard them.

Changing the FTP Timeout

Use the **ftp-timeout** command in the service host submode to change the number of seconds that the FTP client waits before timing out when the sensor is communicating with an FTP server. The default is 300 seconds.

Note You can use the FTP client for downloading updates and configuration files from your FTP server.

To change the FTP timeout, follow these steps:

Step 1 Log in to the sensor using an account with administrator privileges.

Step 2 Enter network settings mode:

```
sensor# configure terminal
sensor(config)# service host
sensor(config-hos)# network-settings
```

Step 3 Change the number of seconds of the FTP timeout:

```
sensor(config-hos-net)# ftp-timeout 500
```

Step 4 Verify the FTP timeout change:

```
sensor(config-hos-net)# show settings
   network-settings
   ---------------------------------------------
      host-ip: 10.89.130.108/23,10.89.130.1
      default: 10.1.9.201/24,10.1.9.1
      host-name: sensor default: sensor
      telnet-option: enabled default: disabled
      access-list (min: 0, max: 512, current: 1)
       ---------------------------------------------
         network-address: 0.0.0.0/0
         ---------------------------------------------
    ---------------------------------------------
      ftp-timeout: 500 seconds default: 300
      login-banner-text: <defaulted>
   ---------------------------------------------
sensor(config-hos-net)#
```

Step 5 Change the value back to the default:

```
sensor(config-hos-net)# default ftp-timeout
```

Step 6 Verify the value has been set back to the default:

```
sensor(config-hos-net)# show settings
   network-settings
   ----------------------------------------------
      host-ip: 10.89.130.108/23,10.89.130.1
      default: 10.1.9.201/24,10.1.9.1
      host-name: sensor default: sensor
      telnet-option: enabled default: disabled
      access-list (min: 0, max: 512, current: 1)
      ---------------------------------------------
         network-address: 0.0.0.0/0
         ---------------------------------------------
      ---------------------------------------------
      ftp-timeout: 300 seconds <defaulted>
      login-banner-text: <defaulted>
   ----------------------------------------------
sensor(config-hos-net)#
```

Step 7 Exit network settings mode:

```
sensor(config-hos-net)# exit
sensor(config-hos)# exit
Apply Changes:?[yes]:
```

Step 8 Press **Enter** to apply the changes or enter **no** to discard them.

Adding a Login Banner

Use the **login-banner-text** *text_message* command to add a login banner that the user sees during login. There is no default.

When you want to start a new line in your message, press **Ctrl-V Enter**.

To add a login banner, follow these steps:

Step 1 Log in to the sensor using an account with administrator privileges.

Step 2 Enter network settings mode:

```
sensor# configure terminal
sensor(config)# service host
sensor(config-hos)# network-settings
```

Step 3 Add the banner login text:

```
sensor(config-hos-net)# login-banner-text This is the banner login text message.
```

Step 4 Verify the banner login text message:

```
sensor(config-hos-net)# show settings
   network-settings
   ----------------------------------------------
      host-ip: 10.89.130.108/23,10.89.130.1
      default: 10.1.9.201/24,10.1.9.1
      host-name: sensor default: sensor
      telnet-option: enabled default: disabled
```

```
      access-list (min: 0, max: 512, current: 1)
      ----------------------------------------------
         network-address: 0.0.0.0/0
         ---------------------------------------------
    ---------------------------------------------
      ftp-timeout: 300 seconds <defaulted>
      login-banner-text: This is the banner login text message. default:
   ---------------------------------------------
sensor(config-hos-net)#
```

Step 5 To remove the login banner text, use the **no** form of the command:

```
sensor(config-hos-net)# no login-banner-text
```

Step 6 Verify the login text has been removed:

```
sensor(config-hos-net)# show settings
   network-settings
   ---------------------------------------------
      host-ip: 10.89.130.108/23,10.89.130.1
      default: 10.1.9.201/24,10.1.9.1
      host-name: sensor default: sensor
      telnet-option: enabled default: disabled
      access-list (min: 0, max: 512, current: 1)
      ----------------------------------------------
         network-address: 0.0.0.0/0
         ---------------------------------------------
      ---------------------------------------------
      ftp-timeout: 300 seconds <defaulted>
      login-banner-text: default:
   ---------------------------------------------
sensor(config-hos-net)#
```

Step 7 Exit network settings mode:

```
sensor(config-hos-net)# exit
sensor(config-hos)# exit
Apply Changes:?[yes]:
```

Step 8 Press **Enter** to apply the changes or enter **no** to discard them.

Changing Web Server Settings

After you run the **setup** command, you can change the following web server settings: the web server port, whether TLS encryption is being used, and the HTTP server header message.

Note The default web server port is 443 if TLS is enabled and 80 if TLS is disabled.

HTTP is the protocol that web clients use to make requests from web servers. The HTTP specification requires a server to identify itself in each response. Attackers sometimes exploit this protocol feature to perform reconnaissance. If the IPS web server identified itself by providing a predictable response, an attacker might learn that an IPS sensor is present.

We recommend that you not reveal to attackers that you have an IPS sensor. Change the **server-id** to anything that does not reveal any information, especially if your web server is available to the Internet.

For example, if you forward a port through a firewall so you can monitor a sensor remotely, you need to set the **server-id**.

To change the web server settings, follow these steps:

Step 1 Log in to the sensor using an account with administrator privileges.

Step 2 Enter web server mode:

```
sensor# configure terminal
sensor(config)# service web-server
```

Step 3 Change the port number:

```
sensor(config-web)# port 8080
```

If you change the port number from the default of 443 to 8080, you receive the following message:

```
Warning: The web server's listening port number has changed from 443 to 8080. This change
will not take effect until the web server is re-started
```

Step 4 Enable or disable TLS:

```
sensor(config-web)# enable-tls {true | false}
```

If you disable TLS, you receive the following message:

```
Warning: TLS protocol support has been disabled. This change will not take effect until
the web server is re-started.
```

Step 5 Change the HTTP server header:

```
sensor(config-web)# server-id Nothing to see here. Move along.
```

Step 6 Verify the web server changes:

```
sensor(config-web)# show settings
   enable-tls: true default: true
   port: 8001 default: 443
   server-id: Nothing to see here. Move along. default: HTTP/1.1 compliant
sensor(config-web)#
```

Step 7 To revert to the defaults, use the **default** form of the commands:

```
sensor(config-web)# default port
sensor(config-web)# default enable-tls
sensor(config-web)# default server-id
```

Step 8 Verify the defaults have been replaced:

```
sensor(config-web)# show settings
   enable-tls: true <defaulted>
   port: 443 <defaulted>
   server-id: HTTP/1.1 compliant <defaulted>
sensor(config-web)#
```

Step 9 Exit web server submode:

```
sensor(config-web)# exit
Apply Changes:?[yes]:
```

Step 10 Press **Enter** to apply the changes or enter `no` to discard them.

Note If you changed the port or enable TLS settings, you must reset the sensor to make the web server use the new settings.

Configuring User Parameters

The following section explains how to create the service account, create users, change passwords, specify privilege level, and view a list of users. It contains the following topics:

Adding and Removing Users

Use the **username** command to create users on the local system. You can add a new user, set the privilege level—administrator, operator, viewer—and set the password for the new user. Use the **no** form of this command to remove a user from the system. This removes the user from CLI and web access.

Caution The **username** command provides username and password authentication for login purposes only. You cannot use this command to remove a user who is logged in to the system. You cannot use this command to remove yourself from the system.

If you do not specify a password, the system prompts you for one. Use the **password** command to change the password for existing users. Use the **privilege** command to change the privilege for existing users.

The username follows the pattern ^[A-Za-z0-9()+:,_/-]+$, which means the username must start with a letter or number, and can include any letter A to Z (capital or small), any number 0 to 9, - and _, and can contain 1 to 64 characters. A valid password is 8 to 32 characters long. All characters except space are allowed.

You receive the following error messages if you do not create a valid password:

- `Error: setEnableAuthenticationTokenStatus: Failure setting the account's password: it`s WAY too short.`
- `Error: setEnableAuthenticationTokenStatus: Failure setting the account's password: it does not contain enough DIFFERENT characters`

Note You cannot use the **privilege** command to give a user service privileges. If you want to give an existing user service privileges, you must remove that user and then use the **username** command to create the service account. For the procedure, see Creating the Service Account, page 4-13.

Note For IPS 5.0 and later, you can no longer remove the **cisco** account. You can disable it using the **no password cisco** command, but you cannot remove it. To use the **no password cisco** command, there must be another administrator account on the sensor. Removing the **cisco** account through the service account is not supported. If you remove the **cisco** account through the service account, the sensor most likely will not boot up, so to recover the sensor you must reinstall the sensor system image.

To add and remove users, follow these steps:

Step 1 Log in to the CLI using an account with administrator privileges.

Step 2 Enter configuration mode:

```
sensor# configure terminal
```

Step 3 Specify the parameters for the user:

```
sensor(config)# username username password password privilege
administrator/operator/viewer
```

Note The username follows the pattern ^[A-Za-z0-9()+:,_/-]+$, which means the username must start with a letter or number, and can include any letter A to Z (capital or small), any number 0 to 9, - and _, and can contain 1 to 64 characters. A valid password is 8 to 32 characters long. All characters except space are allowed.

For example, to add the user “tester” with a privilege level of administrator and the password “testpassword,” enter the following command:

Note If you do not want to see the password in clear text, wait for the password prompt. Do not enter the password along with the username and privilege.

```
sensor(config)# username tester privilege administrator
Enter Login Password: ************
Re-enter Login Password: ************
sensor(config)#
```

Note If you do not specify a privilege level for the user, the user is assigned the default viewer privilege.

Step 4 Verify that the user has been added:

```
sensor(config)# exit
sensor# show users all
    CLI ID   User        Privilege
*   13491    cisco       administrator
             jsmith      operator
             jtaylor     service
```

```
             jroberts    viewer
sensor#
```

A list of users is displayed.

Step 5 To remove a user, use the **no** form of the command:

```
sensor# configure terminal
sensor(config)# no username jsmith
```

Step 6 Verify that the user has been removed:

```
sensor(config)# exit
sensor# show users all
    CLI ID   User        Privilege
*   13491    cisco       administrator
             jtaylor     service
             jroberts    viewer
sensor#
```

The user `jsmith` has been removed.

Note You cannot use this command to remove yourself from the system

Password Recovery

The following password recovery options exist:

- If another Administrator account exists, the other Administrator can change the password.
- If a Service account exists, you can log in to the service account and switch to user root using the command **su - root**. Use the **password** command to change the CLI Administrator account's password. For example, if the Administrator username is "adminu," the command is **password adminu**. You are prompted to enter the new password twice. For more information, see Creating the Service Account, page 4-13.

You can reimage the sensor using either the recovery partition or a system image file. For more information, see Chapter 17, "Upgrading, Downgrading, and Installing System Images."

Creating the Service Account

You can create a service account for TAC to use during troubleshooting. Although more than one user can have access to the sensor, only one user can have service privileges on a sensor. The service account is for support purposes only.

Caution Do not make modifications to the sensor through the service account except under the direction of TAC. If you use the service account to configure the sensor, your configuration is not supported by TAC. Adding services to the operating system through the service account affects proper performance and functioning of the other IPS services. TAC does not support a sensor on which additional services have been added.

Note The root user's password is synchronized to the service account's password when the service account is created. To gain root access you must log in with the service account and switch to user root with the **su - root** command.

Caution You should carefully consider whether you want to create a service account. The service account provides shell access to the system, which makes the system vulnerable. However, you can use the service account to create a new password if the Administrator password is lost. Analyze your situation to decide if you want a service account existing on the system.

Note For IPS 5.0 and later, you can no longer remove the **cisco** account. You can disable it using the **no password cisco** command, but you cannot remove it. To use the **no password cisco** command, there must be another administrator account on the sensor. Removing the **cisco** account through the service account is not supported. If you remove the **cisco** account through the service account, the sensor most likely will not boot up, so to recover the sensor you must reinstall the sensor system image.

To create the service account, follow these steps:

Step 1 Log in to the CLI using an account with administrator privileges.

Step 2 Enter configuration mode:

```
sensor# configure terminal
```

Step 3 Specify the parameters for the service account:

```
sensor(config)# user username privilege service
```

The username follows the pattern ^[A-Za-z0-9()+:,_/-]+$, which means the username must start with a letter or number, and can include any letter A to Z (capital or small), any number 0 to 9, - and _, and can contain 1 to 64 characters.

Step 4 Specify a password when prompted.

A valid password is 8 to 32 characters long. All characters except space are allowed.

If a service account already exists for this sensor, the following error is displayed and no service account is created:

```
Error: Only one service account allowed in UserAccount document
```

Step 5 Exit configuration mode:

```
sensor(config)# exit
sensor#
```

When you use the service account to log in to the CLI, you receive the following warning:

```
************************ WARNING ****************************************************
UNAUTHORIZED ACCESS TO THIS NETWORK DEVICE IS PROHIBITED. This account is intended to be
used for support and troubleshooting purposes only. Unauthorized modifications are not
supported and will require this device to be reimaged to guarantee proper operation.
*****************************************************************************************
```

Configuring Passwords

Use the **password** command to update the password on the local sensor. You can also use this command to change the password for an existing user or to reset the password for a locked account.

A valid password is 8 to 32 characters long. All characters except space are allowed.

To change the password, follow these steps:

Step 1 To change the password for another user or reset the password for a locked account, follow these steps:

a. Log in to the CLI using an account with administrator privileges.

b. Enter configuration mode:

```
sensor# configure terminal
```

c. Change the password for a specific user:

```
sensor(config)# password tester
Enter New Login Password: ******
Re-enter New Login Password: ******
```

Note This example modifies the password for the user "tester."

Step 2 To change your password, follow these steps:

a. Log in to the CLI.

b. Enter configuration mode:

```
sensor# configure terminal
```

c. Change your password:

```
sensor(config)# password
Enter Old Login Password:************
Enter New Login Password: ************
Re-enter New Login Password: ************
```

Changing User Privilege Levels

Use the **privilege** command to change the privilege level—administrator, operator, viewer—for a user.

Note You cannot use the **privilege** command to give a user service privileges. If you want to give an existing user service privileges, you must remove that user and then use the **username** command to create the service account. There can only be one person with service privileges. For the procedure, see Creating the Service Account, page 4-13.

To change the privilege level for a user, follow these steps:

Step 1 Log in to the CLI using an account with administrator privileges.

Step 2 Verify the current privilege of the user *jsmith*:

```
sensor# show users all
    CLI ID   User       Privilege
*   13491    cisco      administrator
             jsmith     viewer
             operator   operator
             service    service
             viewer     viewer
sensor#
```

Step 3 Change the privilege level from viewer to operator:

```
sensor# configure terminal
sensor(config)# privilege user jsmith operator
Warning: The privilege change does not apply to current CLI sessions. It will be applied
to subsequent logins.
sensor(config)#
```

Step 4 Verify that the user's privilege has been changed:

```
sensor(config)# exit
sensor# show users all

    CLI ID   User       Privilege
*   13491    cisco      administrator
             jsmith     operator
             operator   operator
             service    service
             viewer     viewer
sensor#
```

The privilege of the user `jsmith` has been changed from `viewer` to `operator`.

Step 5 Display your current level of privilege:

```
sensor# show privilege
Current privilege level is administrator
```

Viewing User Status

Use the **show users** command to view information about the username and privilege of all users logged in to the sensor, and all user accounts on the sensor regardless of login status.

An * indicates the current user. If an account is locked, the username is surrounded by parentheses. A locked account means that the user failed to enter the correct password after the configured attempts.

Note The number of concurrent CLI sessions is limited based on platform. IDS-4210, IDS-4215, and NM-CIDS are limited to 3 concurrent sessions. All other platforms allow 10 sessions.

To view user information, follow these steps:

Step 1 Log in to the CLI using an account with administrator privileges.

Step 2 Verify the users logged in to the sensor:

```
sensor# show users
    CLI ID   User       Privilege
*   13491    cisco      administrator
sensor#
```

Step 3 Verify all users:

```
sensor# show users all
    CLI ID    User        Privilege
*   13491     cisco       administrator
    5824     (jsmith)     viewer
    9802     tester       operator
sensor#
```

The account of the user `jsmith` is locked.

Step 4 To unlock jsmith's account, reset the password:

```
sensor# configure terminal
sensor(config)# password jsmith
Enter New Login Password: ******
Re-enter New Login Password: ******
```

Configuring Account Locking

Use the **attemptLimit** *number* command in authentication submode to lock accounts so that users cannot keep trying to log in after a certain number of failed attempts. The default is 0, which indicates unlimited authentication attempts. For security purposes, you should change this number.

To configure account locking, follow these steps:

Step 1 Log in to the sensor using an account with administrator privileges.

Step 2 Enter service authentication mode:

```
sensor# configure terminal
sensor(config)# service authentication
```

Step 3 Set the number of attempts users will have to log in to accounts:

```
sensor(config-aut)# attemptLimit 3
```

Step 4 Check your new setting:

```
sensor(config-aut)# show settings
    attemptLimit: 3 defaulted: 0
sensor(config-aut)#
```

Step 5 To set the value back to the system default setting:

```
sensor(config-aut)# default attemptLimit
```

Step 6 Check that the setting has returned to the default:

```
sensor(config-aut)# show settings
    attemptLimit: 0 <defaulted>
sensor(config-aut)#
```

Step 7 Check to see if any users have locked accounts:

Note When you apply a configuration that contains a non-zero value for attemptLimit, a change is made in the SSH server that may subsequently impact your ability to connect with the sensor. When attemptLimit is non-zero, the SSH server requires the client to support challenge-response authentication. If you experience problems after your SSH client connects but before it prompts for a password, you need to enable challenge-response authentication. Refer to the documentation for your SSH client for instructions.

```
sensor(config-aut)# exit
sensor(config)# exit
sensor# show users all
    CLI ID   User       Privilege
*   1349     cisco      administrator
    5824     (jsmith)   viewer
    9802     tester     operator
```

The account of the user `jsmith` is locked as indicated by the parenthesis.

Step 8 To unlock jsmith's account, reset the password:

```
sensor# configure terminal
sensor(config)# password jsmith
Enter New Login Password: ******
Re-enter New Login Password: ******
```

Configuring Time

This section describes the importance of having a reliable time source for the sensor. It contains the following topics:

Time Sources and the Sensor

The sensor requires a reliable time source. All events (alerts) must have the correct UTC and local time stamp, otherwise, you cannot correctly analyze the logs after an attack. When you initialize the sensor, you set up the time zones and summertime settings. For more information, see Chapter 3, "Initializing the Sensor."

Here is a summary of ways to set the time on sensors:

- For appliances
  - Use the **clock set** command to set the time. This is the default.

    For the procedure, see Manually Setting the Clock, page 13-8.
  - Use NTP

    You can configure the appliance to get its time from an NTP time synchronization source. For the procedure, see Configuring a Cisco Router to be an NTP Server, page 4-29. You will need the NTP server IP address, the NTP key ID, and the NTP key value. You can set up NTP on the appliance during initialization or you can configure NTP through the CLI, IDM, or ASDM.

    Note We recommend that you use an NTP time synchronization source.

- For IDSM-2
  - The IDSM-2 can automatically synchronize its clock with the switch time. This is the default.

    Note The UTC time is synchronized between the switch and the IDSM-2. The time zone and summertime settings are not synchronized between the switch and the IDSM-2.

Caution Be sure to set the time zone and summertime settings on both the switch and IDSM-2 to ensure that the UTC time settings are correct. IDSM-2's local time could be incorrect if the time zone and/or summertime settings do not match between IDSM-2 and the switch. For more information, see Synchronizing IPS Module System Clocks with Parent Device System Clocks, page 4-20.

  - Use NTP

    You can configure IDSM-2 to get its time from an NTP time synchronization source. For the procedure, see Configuring a Cisco Router to be an NTP Server, page 4-29. You will need the NTP server IP address, the NTP key ID, and the NTP key value. You can configure IDSM-2 to use NTP during initialization or you can set up NTP through the CLI, IDM, or ASDM.

    Note We recommend that you use an NTP time synchronization source.

- For NM-CIDS
  - NM-CIDS can automatically synchronize its clock with the clock in the router chassis in which it is installed (parent router). This is the default.

    Note The UTC time is synchronized between the parent router and NM-CIDS. The time zone and summertime settings are not synchronized between the parent router and NM-CIDS.

Caution Be sure to set the time zone and summertime settings on both the parent router and NM-CIDS to ensure that the UTC time settings are correct. NM-CIDS's local time could be incorrect if the time zone and/or summertime settings do not match between NM-CIDS and the router. For more information, see Synchronizing IPS Module System Clocks with Parent Device System Clocks, page 4-20.

– Use NTP

You can configure NM-CIDS to get its time from an NTP time synchronization source, such as a Cisco router other than the parent router. For the procedure, see Configuring a Cisco Router to be an NTP Server, page 4-29. You will need the NTP server IP address, the NTP key ID, and the NTP key value. You can configure NM-CIDS to use NTP during initialization or you can set up NTP through the CLI, IDM, or ASDM.

Note We recommend that you use an NTP time synchronization source.

- For AIP-SSM:
  – AIP-SSM can automatically synchronize its clock with the clock in the ASA in which it is installed. This is the default.

Note The UTC time is synchronized between ASA and AIP-SSM. The time zone and summertime settings are not synchronized between ASA and AIP-SSM.

Caution Be sure to set the time zone and summertime settings on both ASA and AIP-SSM to ensure that the UTC time settings are correct. AIP-SSM's local time could be incorrect if the time zone and/or summertime settings do not match between AIP-SSM and ASA. For more information, see Synchronizing IPS Module System Clocks with Parent Device System Clocks, page 4-20.

– Use NTP

You can configure AIP-SSM to get its time from an NTP time synchronization source, such as a Cisco router other than the parent router. For the procedure, see Configuring a Cisco Router to be an NTP Server, page 4-29. You will need the NTP server IP address, the NTP key ID, and the NTP key value. You can configure AIP-SSM to use NTP during initialization or you can set up NTP through the CLI, IDM, or ASDM.

Note We recommend that you use an NTP time synchronization source.

Synchronizing IPS Module System Clocks with Parent Device System Clocks

All IPS modules (IDSM-2, NM-CIDS, and AIP-SSM) synchronize their system clocks to the parent chassis clock (switch, router, or firewall) each time the module boots up and any time the parent chassis clock is set. The module clock and parent chassis clock tend to drift apart over time. The difference can be as much as several seconds per day. To avoid this problem, make sure that both the module clock and the parent clock are synchronized to an external NTP server. If only the module clock or only the parent chassis clock is synchronized to an NTP server, the time drift occurs. For more information on NTP, see Configuring NTP, page 4-29.

Verifying the Sensor is Synchronized with the NTP Server

In IPS 5.1, you cannot apply an incorrect NTP configuration, such as an invalid NTP key value or ID, to the sensor. If you try to apply an incorrect configuration, you receive an error message. To verify the NTP configuration, use the **show statistics host** command to gather sensor statistics. The NTP statistics section provides NTP statistics including feedback on sensor synchronization with the NTP server.

To verify the NTP configuration, follow these steps:

Step 1 Log in to the sensor.

Step 2 Generate the host statistics:

```
sensor# show statistics host
     ...
     NTP Statistics
             remote           refid      st t when poll reach   delay   offset  jitter
         11.22.33.44     CHU_AUDIO(1)     8 u   36   64    1    0.536    0.069   0.001
         LOCAL(0)        73.78.73.84      5 l   35   64    1    0.000    0.000   0.001
        ind assID status  conf reach auth condition  last_event cnt
          1 10372  f014   yes   yes   ok     reject   reachable  1
          2 10373  9014   yes   yes  none    reject   reachable  1
        status = Not Synchronized
     ...
```

Step 3 Generate the hosts statistics again after a few minutes:

```
sensor# show statistics host
     ...
     NTP Statistics
             remote           refid      st t when poll reach   delay   offset  jitter
        *11.22.33.44     CHU_AUDIO(1)     8 u   22   64  377    0.518   37.975  33.465
         LOCAL(0)        73.78.73.84      5 l   22   64  377    0.000    0.000   0.001
        ind assID status  conf reach auth condition  last_event cnt
          1 10372  f624   yes   yes   ok   sys.peer   reachable  2
          2 10373  9024   yes   yes  none    reject   reachable  2
        status = Synchronized
```

Step 4 If the status continues to read `Not Synchronized`, check with the NTP server administrator to make sure the NTP server is configured correctly.

Correcting Time on the Sensor

If you set the time incorrectly, your stored events will have the incorrect time because they are stamped with the time the event was created.

The Event Store time stamp is always based on UTC time. If during the original sensor setup, you set the time incorrectly by specifying 8:00 p.m. rather than 8:00 a.m., when you do correct the error, the corrected time will be set backwards. New events might have times older than old events.

For example, if during the initial setup, you configure the sensor as central time with daylight saving time enabled and the local time is 8:04 p.m., the time is displayed as 20:04:37 CDT and has an offset from UTC of -5 hours (01:04:37 UTC, the next day). A week later at 9:00 a.m., you discover the error: the clock shows 21:00:23 CDT. You then change the time to 9:00 a.m. and now the clock shows 09:01:33 CDT. Because the offset from UTC has not changed, it requires that the UTC time now be 14:01:33 UTC, which creates the time stamp problem.

To ensure the integrity of the time stamp on the event records, you must clear the event archive of the older events by using the **clear events** command. For more information on the **clear events** command, see Configuring a Cisco Router to be an NTP Server, page 4-29.

Caution You cannot remove individual events.

Configuring Time on the Sensor

This section describes how to configure time on the sensor so that your events are time-stamped correctly. It contains the following topics:

System Clock

This section describes how to display and manually set the system clock on the appliance. It contains the following topics:

Displaying the System Clock

Use the **show clock** [**detail**] command to display the system clock. You can use the **detail** option to indicate the clock source (NTP or system) and the current summertime setting (if any).

The system clock keeps an authoritative flag that indicates whether the time is authoritative (believed to be accurate). If the system clock has been set by a timing source, such as NTP, the flag is set.

Symbol	Description
*	Time is not authoritative.
(blank)	Time is authoritative.
.	Time is authoritative, but NTP is not synchronized.

To display the system clock, follow these steps:

Step 1 Log in to the CLI.

Step 2 Display the system clock:

```
sensor# show clock
22:39:21 UTC Sat Jan 25 2003
```

Step 3 Display the system clock with details:

```
sensor# show clock detail
22:39:21 CST Sat Jan 25 2003
Time source is NTP
```

```
Summer time starts 02:00:00 CST Sun Apr 7 2004
Summer time ends 02:00:00 CDT Sun Oct 27 2004
```

This indicates that the sensor is getting its time from NTP and that is configured and synchronized.

```
sensor# show clock detail
*12:19:22 CST Sat Dec 04 2004
No time source
Summer time starts 02:00:00 CST Sun Apr 7 2004
Summer time ends 02:00:00 CDT Sun Oct 27 2004
```

This indicates that no time source is configured.

Manually Setting the System Clock

Use the **clock set** *hh:mm [:ss] month day year* command to manually set the clock on the appliance. Use this command if no other time sources are available.

Note You do not need to set the system clock if your sensor is synchronized by a valid outside timing mechanism such as an NTP clock source.

For the procedure for configuring NTP, see Configuring NTP, page 4-29. For an explanation of the importance of having a valid time source for the sensor, see Time Sources and the Sensor, page 4-18. For an explanation of what to do if you set the clock incorrectly, see Correcting Time on the Sensor, page 4-21.

The **clock set** command does not apply to the following platforms:

- IDSM-2
- NM-CIDS
- AIP-SSM-10
- AIP-SSM-20

To manually set the clock on the appliance, follow these steps:

Step 1 Log in to the CLI using an account with administrator privileges.

Step 2 Set the clock manually:

```
sensor# clock set 13:21 July 29 2004
```

Note The time format is 24-hour time.

Configuring Summertime Settings

You can configure summertime settings if you did not do so during initialization of the sensor. Or you can change them after initialization.

Note Summertime is a term for daylight saving time.

This section contains the following topics:

- Configuring Recurring Summertime Settings, page 4-24
- Configuring Non-recurring Summertime Settings, page 4-26

Configuring Recurring Summertime Settings

Use the **summertime-option recurring** command to configure the sensor to switch to summertime settings on a recurring basis. The default is recurring.

To configure the sensor to switch to summertime settings on a recurring basis, follow these steps:

Step 1 Log in to the sensor using an account with administrator privileges.

Step 2 Enter summertime recurring submode:

```
sensor# configure terminal
sensor(config)# service host
sensor(config-hos)# summertime-option recurring
```

Step 3 Enter start summertime submode:

```
sensor(config-hos-rec)# start-summertime
```

Step 4 Configure the start summertime parameters:

a. Enter the day of the week you want to start summertime settings:

```
sensor(config-hos-rec-sta)# day-of-week monday
```

b. Enter the month you want to start summertime settings:

```
sensor(config-hos-rec-sta)# month april
```

c. Enter the time of day you want to start summertime settings:

```
sensor(config-hos-rec-sta)# time-of-day 12:00:00
```

The format is hh:mm:ss.

d. Enter the week of the month you want to start summertime settings:

```
sensor(config-hos-rec-sta)# week-of-month first
```

The values are first through fifth, or last.

e. Verify your settings:

```
sensor(config-hos-rec-sta)# show settings
   start-summertime
   ---------------------------------------------
      month: april default: april
      week-of-month: first default: first
      day-of-week: monday default: sunday
      time-of-day: 12:00:00 default: 02:00:00
   ---------------------------------------------
sensor(config-hos-rec-sta)#
```

Step 5 Enter end summertime submode:

```
sensor(config-hos-rec-sta)# exit
```

```
sensor(config-hos-rec)# end-summertime
```

Step 6 Configure the end summertime parameters:

a. Enter the day of the week you want to end summertime settings:

```
sensor(config-hos-rec-end)# day-of-week friday
```

b. Enter the month you want to end summertime settings:

```
sensor(config-hos-rec-end)# month october
```

c. Enter the time of day you want to end summertime settings:

```
sensor(config-hos-rec-end)# time-of-day 05:15:00
```

The format is hh:mm:ss.

d. Enter the week of the month you want to end summertime settings:

```
sensor(config-hos-rec-end)# week-of-month last
```

The values are first through fifth, or last.

e. Verify your settings:

```
sensor(config-hos-rec-end)# show settings
   end-summertime
   ----------------------------------------------
      month: october default: october
      week-of-month: last default: last
      day-of-week: friday default: sunday
      time-of-day: 05:15:00 default: 02:00:00
   ----------------------------------------------
sensor(config-hos-rec-end)#
```

Step 7 Specify the local time zone used during summertime:

```
sensor(config-hos-rec-end)# exit
sensor(config-hos-rec)# summertime-zone-name CDT
```

Step 8 Specify the offset:

```
sensor(config-hos-rec)# offset 60
```

Note Changing the time zone offset requires the sensor to reboot.

Step 9 Verify your settings:

```
sensor(config-hos-rec)# show settings
   recurring
   ----------------------------------------------
      offset: 60 minutes default: 60
      summertime-zone-name: CDT
      start-summertime
      ----------------------------------------------
         month: april default: april
         week-of-month: first default: first
         day-of-week: monday default: sunday
         time-of-day: 12:00:00 default: 02:00:00
      ----------------------------------------------
      end-summertime
      ----------------------------------------------
         month: october default: october
         week-of-month: last default: last
```

```
          day-of-week: friday default: sunday
          time-of-day: 05:15:00 default: 02:00:00
        ---------------------------------------------
    ---------------------------------------------
```

Step 10 Exit recurring summertime submode:

```
sensor(config-hos-rec)# exit
sensor(config-hos)# exit
Apply Changes:?[yes]:
```

Step 11 Press **Enter** to apply the changes or enter **no** to discard them.

Configuring Non-recurring Summertime Settings

Use the **summertime-option non-recurring** command to configure the sensor to switch to summer time settings on a one-time basis. The default is recurring.

To configure the sensor to switch to summertime settings on a one-time basis, follow these steps:

Step 1 Log in to the sensor using an account with administrator privileges.

Step 2 Enter summertime non-recurring submode:

```
sensor# configure terminal
sensor(config)# service host
sensor(config-hos)# summertime-option non-recurring
```

Step 3 Enter start summertime submode:

```
sensor(config-hos-non)# start-summertime
```

Step 4 Configure the start summertime parameters:

a. Enter the date you want to start summertime settings:

```
sensor(config-hos-non-sta)# date 2004-05-15
```

The format is yyyy-mm-dd.

b. Enter the time you want to start summertime settings:

```
sensor(config-hos-non-sta)# time 12:00:00
```

The format is hh:mm:ss.

c. Verify your settings:

```
sensor(config-hos-non-sta)# show settings
   start-summertime
   ---------------------------------------------
      date: 2004-05-15
      time: 12:00:00
   ---------------------------------------------
sensor(config-hos-non-sta)#
```

Step 5 Enter end summertime submode:

```
sensor(config-hos-non-sta)# exit
sensor(config-hos-non)# end-summertime
```

Step 6 Configure the end summertime parameters:

a. Enter the date you want to end summertime settings:

```
sensor(config-hos-non-end)# date 2004-10-31
```

The format is yyyy-mm-dd.

b. Enter the time you want to end summertime settings:

```
sensor(config-hos-non-end)# time 12:00:00
```

The format is hh:mm:ss.

c. Verify your settings:

```
sensor(config-hos-non-end)# show settings
   end-summertime
   ----------------------------------------------
      date: 2004-10-31
      time: 12:00:00
   ----------------------------------------------
sensor(config-hos-non-end)#
```

Step 7 Specify the local time zone used during summertime:

```
sensor(config-hos-non-end)# exit
sensor(config-hos-non)# summertime-zone-name CDT
```

Step 8 Specify the offset:

```
sensor(config-hos-non)# offset 60
```

Note Changing the time zone offset requires the sensor to reboot.

Step 9 Verify your settings:

```
sensor(config-hos-non)# show settings
   non-recurring
   ----------------------------------------------
      offset: 60 minutes default: 60
      summertime-zone-name: CDT
      start-summertime
      ----------------------------------------------
         date: 2004-05-15
         time: 12:00:00
      ----------------------------------------------
      end-summertime
      ----------------------------------------------
         date: 2004-10-31
         time: 12:00:00
      ----------------------------------------------
   ----------------------------------------------
sensor(config-hos-non)#
```

Step 10 Exit non-recurring summertime submode:

```
sensor(config-hos-non)# exit
sensor(config-hos)# exit
Apply Changes:?[yes]:
```

Step 11 Press **Enter** to apply the changes or enter **no** to discard them.

Configuring Timezones Settings

Use the **time-zone-settings** command to configure the timezone settings on the sensor, such as the timezone name the sensor displays whenever summertime settings are not in effect and the offset.

To configure the timezone settings on the sensor, follow these steps:

Step 1 Log in to the sensor using an account with administrator privileges.

Step 2 Enter timezone settings submode:

```
sensor# configure terminal
sensor(config)# service host
sensor(config-hos)# time-zone-settings
```

Step 3 Configure the timezone name that is displayed whenever summertime settings are not in effect:

The default is UTC.

```
sensor(config-hos-tim)# standard-time-zone-name CST
```

Step 4 Configure the offset in minutes:

```
sensor(config-hos-tim)# offset -360
```

The offset is the number of minutes you add to UTC to get the local time. The default is 0.

Note Changing the time zone offset requires the sensor to reboot.

Step 5 Verify your settings:

```
sensor(config-hos-tim)# show settings
   time-zone-settings
   ----------------------------------------------
      offset: -360 minutes default: 0
      standard-time-zone-name: CST default: UTC
   ----------------------------------------------
sensor(config-hos-tim)#
```

Step 6 Exit timezone settings submode:

```
sensor(config-hos-tim)# exit
sensor(config-hos)# exit
Apply Changes:?[yes]:
```

Step 7 Press **Enter** to apply the changes or enter **no** to discard them.

Configuring NTP

This section describes how to configure a Cisco router to be an NTP server and how to configure the sensor to use an NTP server as its time source. It contains the following topics:

- Configuring a Cisco Router to be an NTP Server, page 4-29
- Configuring the Sensor to Use an NTP Time Source, page 4-30

Configuring a Cisco Router to be an NTP Server

The sensor requires an authenticated connection with an NTP server if it is going to use the NTP server as its time source. The sensor supports only the MD5 hash algorithm for key encryption. Use the following procedure to activate a Cisco router to act as an NTP server and use its internal clock as the time source.

Caution The sensor NTP capability is designed to be compatible with Cisco routers acting as NTP servers. The sensor may work with other NTP servers, but is not tested or supported.

Note Remember the NTP server's key ID and key values. You will need them along with the IP address of the NTP server when you configure the sensor to use the NTP server as its time source. For this procedure, see Configuring the Sensor to Use an NTP Time Source, page 4-30.

To set up a Cisco router to act as an NTP server, follow these steps:

Step 1 Log in to the router.

Step 2 Enter configuration mode:

```
router# configure terminal
```

Step 3 Create the key ID and key value:

```
router(config)# ntp authentication-key key_ID md5 key_value
```

The key ID can be a number between 1 and 65535. The key value is text (numeric or character). It is encrypted later.

Example:

```
router(config)# ntp authentication-key 100 md5 attack
```

Note The sensor only supports MD5 keys.

Note Keys may already exist on the router. Use the **show running configuration** command to check for other keys. You can use those values for the trusted key in Step 4.

Step 4 Designate the key you just created in Step 3 as the trusted key (or use an existing key):

```
router(config)# ntp trusted-key key_ID
```

The trusted key ID is the same number as the key ID in Step 3.

Example:

```
router(config)# ntp trusted-key 100
```

Step 5 Specify the interface on the router that the sensor will communicate with:

```
router(config)# ntp source interface_name
```

Example:

```
router(config)# ntp source FastEthernet 1/0
```

Step 6 Specify the NTP master stratum number to be assigned to the sensor:

```
router(config)# ntp master stratum_number
```

Example:

```
router(config)# ntp master 6
```

The NTP master stratum number identifies the server's relative position in the NTP hierarchy. You can choose a number between 1 and 15. It is not important to the sensor which number you choose.

Configuring the Sensor to Use an NTP Time Source

The sensor requires a consistent time source. We recommend that you use an NTP server. Use the following procedure to configure the sensor to use the NTP server as its time source.

Caution The sensor NTP capability is designed to be compatible with Cisco routers acting as NTP servers. The sensor may work with other NTP servers, but is not tested or supported.

Note You must obtain the NTP server IP address, NTP server key ID, and the key value from the NTP server. For more information, see Configuring a Cisco Router to be an NTP Server, page 4-29.

To configure the sensor to use an NTP server as its time source, follow these steps:

Step 1 Log in to the CLI using an account with administrator privileges.

Step 2 Enter configuration mode:

```
sensor# configure terminal
```

Step 3 Enter service host mode:

```
sensor(config)# service host
```

Step 4 Enter NTP configuration mode:

```
sensor(config-hos)# ntp-option enable
```

Step 5 Enter the NTP server IP address and key ID:

```
sensor(config-hos-ena)# ntp-servers ip_address key-id key_ID
```

The key ID is a number between 1 and 65535. This is the key ID that you already set up on the NTP server. See Step 3 of Configuring a Cisco Router to be an NTP Server, page 4-29.

Example:

```
sensor(config-hos-ena)# ntp-servers 10.16.0.0 key-id 100
```

Step 6 Enter the NTP server's key value:

```
sensor(config-hos-ena)# ntp-keys key_ID md5-key key_value
```

The key value is text (numeric or character). This is the key value that you already set up on the NTP server. See Step 3 of Configuring a Cisco Router to be an NTP Server, page 4-29.

Example:

```
sensor(config-hos-ena)# ntp-keys 100 md5-key attack
```

Step 7 Verify the NTP settings:

```
sensor(config-hos-ena)# show settings
   enabled
   ----------------------------------------------
      ntp-keys (min: 1, max: 1, current: 1)
      ----------------------------------------------
         key-id: 100
         ----------------------------------------------
            md5-key: attack
         ----------------------------------------------
      ----------------------------------------------
      ntp-servers (min: 1, max: 1, current: 1)
      ----------------------------------------------
         ip-address: 10.16.0.0
         key-id: 100
      ----------------------------------------------
----------------------------------------------
sensor(config-hos-ena)#
```

Step 8 Exit NTP configuration mode:

```
sensor(config-hos-ena)# exit
sensor(config-hos)# exit
Apply Changes:?[yes]
```

Step 9 Press **Enter** to apply the changes or enter **no** to discard them.

Configuring SSH

This section describes SSH on the sensor, and contains the following topics:

Understanding SSH

SSH provides strong authentication and secure communications over channels that are not secure.

SSH encrypts your connection to the sensor and provides a key so you can validate that you are connecting to the correct sensor. SSH also provides authenticated and encrypted access to other devices that the sensor connects to for blocking.

SSH authenticates the hosts or networks using one or more of the following:

- Password
- User RSA public key

SSH protects against the following:

- IP spoofing—A remote host sends out packets pretending to come from another trusted host.

  SSH even protects against a spoofer on the local network who can pretend he is your router to the outside.
- IP source routing—A host pretends an IP packet comes from another trusted host.
- DNS spoofing—An attacker forges name server records.
- Interception of clear text passwords and other data by intermediate hosts.
- Manipulation of data by those in control of intermediate hosts.
- Attacks based on listening to X authentication data and spoofed connection to the X11 server.

Note SSH never sends passwords in clear text.

Adding Hosts to the Known Hosts List

You must add hosts to the SSH known hosts list so that the sensor can recognize the hosts that it can communicate with through SSH. These hosts are SSH servers that the sensor needs to connect to for upgrades and file copying, and other hosts, such as Cisco routers, PIX Firewalls, and Catalyst switches that the sensor will connect to for blocking.

Use the **ssh host-key** *ip-address* [*key-modulus-length public-exponent public-modulus*] command to add an entry to the known hosts list. If you do not know the values for the modulus, exponent, and length, the system displays the MD5 fingerprint and bubble babble for the requested IP address. You can then select to add the key to the list.

Caution When you use the **ssh host-key** *ip-address* command, the SSH server at the specified IP address is contacted to obtain the required key over the network. The specified host must by accessible at the moment the command is issued. If the host is unreachable, you must use the full form of the command, **ssh host-key** *ip-address* [*key-modulus-length public-exponent public-modulus*], to confirm the fingerprint of the key displayed to protect yourself from accepting an attacker's key.

Note To modify a key for an IP address, the entry must be removed and recreated. Use the **no** form of the command to remove the entry.

To add a host to the SSH known hosts list, follow these steps:

Step 1 Log in to the CLI using an account with administrator or operator privileges.

Step 2 Enter configuration mode:

```
sensor# configure terminal
```

Step 3 Add an entry to the known hosts list:

```
sensor(config)# ssh host-key 10.16.0.0
MD5 fingerprint is F3:10:3E:BA:1E:AB:88:F8:F5:56:D3:A6:63:42:1C:11
Bubble Babble is xucis-hehon-kizog-nedeg-zunom-kolyn-syzec-zasyk-symuf-rykum-sexyx
Would you like to add this to the known hosts table for this host?[yes]
```

The MD5 fingerprint appears. You are prompted to add it to the known hosts list:

If the host is not accessible when the command is issued, the following message appears:

```
Error: getHostSshKey: socket connect failed [4,111]
```

Step 4 Enter **yes** to have the fingerprint added to the known hosts list.

Step 5 Verify that the host was added:

```
sensor(config)# exit
sensor# show ssh host-keys
10.89.146.110
```

Step 6 View the key for a specific IP address:

```
sensor# show ssh host-keys 10.16.0.0
1024 35
139306213541835240385332922253968814685684523520064131997839905113640120217816869696708721
704631322844292073851730565044879082670677554157937058485203995572114631296604552161309712
601068614812749969593513740598331393154884988302302182922353335152653860589163651944997842
874583627883277460138506084043415861927
MD5: 49:3F:FD:62:26:58:94:A3:E9:88:EF:92:5F:52:6E:7B
Bubble Babble: xebiz-vykyk-fekuh-rukuh-cabaz-paret-gosym-serum-korus-fypop-huxyx
sensor#
```

Step 7 Remove an entry:

```
sensor(config)# no ssh host-key 10.16.0.0
```

The host is removed from the SSH known hosts list.

Step 8 Verify the host was removed:

```
sensor(config)# exit
sensor# show ssh host-keys
```

The IP address no longer appears in the list.

Adding SSH Authorized Public Keys

Use the **ssh authorized-key** command to define public keys for a client allowed to use RSA authentication to log in to the local SSH server.

The following options apply:

- *id*—1 to 256-character string that uniquely identifies the authorized key. You can use numbers, "_," and "-," but spaces and "?" are not acceptable.

- *key-modulus-length*—An ASCCI decimal integer in the range[511, 2048].
- *public-exponent*—An ASCII decimal integer in the range [3, 2^32].
- *public-modulus*—An ASCII decimal integer, *x*, such that (2^(key-modulus-length-1)) < x < (2^(key-modulus-length)).

Each user who can log in to the sensor has a list of authorized public keys. An SSH client with access to any of the corresponding RSA private keys can log in to the sensor as the user without entering a password.

Use an RSA key generation tool on the client where the private key is going to reside. Then, display the generated public key as a set of three numbers (modulus length, public exponent, public modulus) and enter those numbers as parameters for the **ssh authorized-key** command.

Note You configure your own list of SSH authorized keys. An administrator cannot manage the list of SSH authorized keys for other users on the sensor.

Note An SSH authorized key provides better security than passwords if the private key is adequately safeguarded. The best practice is to create the private key on the same host where it will be used and store it with a passphrase on a local file system. To minimize password or passphrase prompts, use a key agent.

Note To modify an authorized key, you must remove and recreate the entry. Use the **no** form of the command to remove the entry. Users can only create and remove their own keys.

To add a key entry to the SSH authorized keys list for the current user, follow these steps:

Step 1 Log in to the CLI.

Step 2 Add a key to the authorized keys list for the current user:

```
sensor# configure terminal
sensor(config)# ssh authorized-key system1 1023 37
660222729556609833380897067163729433570828686860008172017802434921804214207813035920829509101701358480525039993932112503147452768378620911189986653716089813147922086044739911341369642870682319361928148521864094557416306138786468335115835910404940213136954353396163449793497050167925831465486221464674219970 57
sensor(config)#
```

Step 3 Verify that the key was added:

```
sensor(config)# exit
sensor# show ssh authorized-keys
system1
sensor#
```

Step 4 View the key for a specific ID:

```
sensor# show ssh authorized-keys system1
1023 37 660222729556609833380897067163729433570828686860008172017802434921804214
207813035920829509101701358480525039993932112503147452768378620911189986653716 08
981314792208604473991134136964287068231936192814852186409455741630613878646833 51
158359104049402131369543533961634497934970501679258314654862214646742199705 7
sensor#
```

Step 5 Remove an entry from the list of SSH authorized keys:

```
sensor# configure terminal
sensor(config)# no ssh authorized-key system1
```

The key is removed from the SSH authorized keys list.

Step 6 Verify the entry was removed:

```
sensor(config)# exit
sensor# show ssh authorized-keys
```

The key system1 no longer appears in the list:

If you enter the former ID, you receive an error message:

```
sensor# show ssh authorized-keys system1
Error: Requested id does not exist for the current user.
sensor#
```

Generating a New SSH Server Key

Use the **ssh generate-key** command to change the SSH server host key. The displayed fingerprint matches the one displayed in the remote SSH client in future connections with this sensor if the remote client is using SSH 1.5.

To generate a new SSH server host key, follow these steps:

Step 1 Log in to the CLI using an account with administrator privileges.

Step 2 Generate the new server host key:

```
sensor# ssh generate-key
MD5: 93:F5:51:58:C7:FD:40:8C:07:26:5E:29:13:C8:33:AE
Bubble Babble: ximal-sudez-kusot-gosym-levag-fegoc-holez-cakar-kunel-nylis-kyxox
sensor#
```

Caution The new key replaces the existing key, which requires you to update the known hosts tables on remote systems with the new host key so that future connections succeed. You can update the known hosts tables on remote systems using the **ssh host-key** command. For the procedure, see Adding Hosts to the Known Hosts List, page 4-32.

Step 3 Display the current SSH server host key:

```
sensor# show ssh server-key
1024 35
137196765426571419509124895787229630062726389801071715581921573847280637533000158590028798
074385824867184332364758899959675370523879609376174812179228415215782949029183962207840731
771645803509837259475421477212459797170806510716077556010753169312675023860474987441651041
217710152766990480431898217878170000647
MD5: 93:F5:51:58:C7:FD:40:8C:07:26:5E:29:13:C8:33:AE
Bubble Babble: ximal-sudez-kusot-gosym-levag-fegoc-holez-cakar-kunel-nylis-kyxox
sensor#
```

Configuring TLS

This section describes TLS on the sensor, and contains the following topics:

Understanding TLS

IPS 5.1 contains a web server that is running IDM and that the management stations, such as VMS, connect to. Blocking forwarding sensors also connect to the web server of the master blocking sensor. To provide security, this web server uses an encryption protocol known as TLS, which is closely related to SSL protocol. When you enter a URL into the web browser that starts with `https://`*ip_address*, the web browser responds by using either TLS or SSL protocol to negotiate an encrypted session with the host.

Caution The web browser initially rejects the certificate presented by IDM because it does not trust the CA.

Note IDM is enabled by default to use TLS and SSL. We highly recommend that you use TLS and SSL.

The process of negotiating an encrypted session in TLS is called "handshaking," because it involves a number of coordinated exchanges between client and server. The server sends its certificate to the client. The client performs the following three-part test on this certificate:

1. Is the issuer identified in the certificate trusted?

   Every web browser ships with a list of trusted third-party CAs. If the issuer identified in the certificate is among the list of CAs trusted by your browser, the first test is passed.

2. Is the date within the range of dates during which the certificate is considered valid?

   Each certificate contains a Validity field, which is a pair of dates. If the date falls within this range of dates, the second test is passed.

3. Does the common name of the subject identified in the certificate match the URL hostname?

   The URL hostname is compared with the subject common name. If they match, the third test is passed.

When you direct your web browser to connect with IDM, the certificate that is returned fails because the sensor issues its own certificate (the sensor is its own CA) and the sensor is not already in the list of CAs trusted by your browser.

When you receive an error message from your browser, you have three options:

- Disconnect from the site immediately.
- Accept the certificate for the remainder of the web browsing session.
- Add the issuer identified in the certificate to the list of trusted CAs of the web browser and trust the certificate until it expires.

The most convenient option is to permanently trust the issuer. However, before you add the issuer, use out-of-band methods to examine the fingerprint of the certificate. This prevents you from being victimized by an attacker posing as a sensor. Confirm that the fingerprint of the certificate appearing in your web browser is the same as the one on your sensor.

Caution If you change the organization name or hostname of the sensor, a new certificate is generated the next time the sensor is rebooted. The next time your web browser connects to IDM, you will receive the manual override dialog boxes. You must perform the certificate fingerprint validation again for Internet Explorer, Netscape, and Mozilla.

Adding TLS Trusted Hosts

In certain situations, the sensor uses TLS/SSL to protect a session it establishes with a remote web server. For these sessions to be secure from man-in-the-middle attacks you must establish trust of the remote web servers' TLS certificates. A copy of the TLS certificate of each trusted remote host is stored in the trusted hosts list.

Use the **tls trusted-host ip-address** *ip-address* [**port** *port*] command to add a trusted host to the trusted hosts list. This command retrieves the TLS certificate from the specified host/port and displays its fingerprint. You can accept or reject the fingerprint based on information retrieved directly from the host you are requesting to add. The default port is 443.

Each certificate is stored with an identifier field (**id**). For the IP address and default port, the identifier field is **ipaddress**. For the IP address and specified port, the identifier field is **ipaddress:port**.

Caution TLS at the specified IP address is contacted to obtain the required fingerprint over the network. The specified host must by accessible at the moment the command is issued. Use an alternate method to confirm the fingerprint to protect yourself from accepting an attacker's certificate

To add a trusted host to the trusted hosts list, follow these steps:

Step 1 Log in to the CLI using an account with administrator or operator privileges.

Step 2 Add the trusted host:

```
sensor# configure terminal
sensor(config)# tls trusted-host ip-address 10.16.0.0
Certificate MD5 fingerprint is 4F:BA:15:67:D3:E6:FB:51:8A:C4:57:93:4D:F2:83:FE
Certificate SHA1 fingerprint is B1:6F:F5:DA:F3:7A:FB:FB:93:E9:2D:39:B9:99:08:D4:
47:02:F6:12
Would you like to add this to the trusted certificate table for this host?[yes]:
```

The MD5 and SHA1 fingerprints appear. You are prompted to add the trusted host.

If the connection cannot be established, the transaction fails:

```
sensor(config)# tls trusted-host ip-address 10.89.146.110 port 8000
Error: getHostCertificate: socket connect failed [4,111]
```

Step 3 Enter **yes** to accept the fingerprint.

```
Certificate ID: 10.89.146.110 successfully added to the TLS trusted host table.
sensor(config)#
```

The host has been added to the TLS trusted host list. The Certificate ID stored for the requested certificate is displayed when the command is successful.

Step 4 Verify that the host was added:

```
sensor(config)# exit
sensor# show tls trusted-hosts
10.89.146.110
sensor#
```

Step 5 View the fingerprint for a specific host:

```
sensor# show tls trusted-hosts 10.89.146.110
MD5: 4F:BA:15:67:D3:E6:FB:51:8A:C4:57:93:4D:F2:83:FE
SHA1: B1:6F:F5:DA:F3:7A:FB:FB:93:E9:2D:39:B9:99:08:D4:47:02:F6:12
sensor#
```

Step 6 Remove an entry from the trusted hosts list:

```
sensor# configure terminal
sensor(config)# no tls trusted-host 10.89.146.110
```

The host is removed from the trusted hosts list.

Step 7 Verify the entry was removed from the trusted host list:

```
sensor(config)# exit
sensor# show tls trusted-hosts
No entries
```

The IP address no longer appears in the list:

Displaying and Generating the Server Certificate

A TLS certificate is generated when the sensor is first started. Use the **tls generate-key** command to generate a new server self-signed X.509 certificate.

Note The sensor's IP address is included in the certificate. If you change the sensor's IP address, the sensor automatically generates a new certificate.

Caution The new certificate replaces the existing certificate, which requires you to update the trusted hosts lists on remote systems with the new certificate so that future connections succeed. You can update the trusted hosts lists on remote IPS sensors using the **tls trusted-host** command. For the procedure, see Adding TLS Trusted Hosts, page 4-37. If the sensor is a master blocking sensor, you must update the trusted hosts lists on the remote sensors that are sending block requests to the master blocking sensor.

To generate a new TLS certificate, follow these steps:

Step 1 Log in to the CLI using an account with administrator privileges.

Step 2 Generate the new certificate:

```
sensor# tls generate-key
MD5 fingerprint is FD:83:6E:41:D3:88:48:1F:44:7F:AF:5D:52:60:89:DE
SHA1 fingerprint is 4A:2B:79:A0:82:8B:65:3A:83:B5:D9:50:C0:8E:F6:C6:B0:30:47:BB
```

```
sensor#
```

Step 3 Verify that the key was generated:

```
sensor# show tls fingerprint
MD5: FD:83:6E:41:D3:88:48:1F:44:7F:AF:5D:52:60:89:DE
SHA1: 4A:2B:79:A0:82:8B:65:3A:83:B5:D9:50:C0:8E:F6:C6:B0:30:47:BB
sensor#
```

Installing the License Key

This section describes the IPS license key and how to install it. It contains the following topics:

Overview

Although the sensor functions without the license key, you must have a license key to obtain signature updates. To obtain a license key, you must have the following:

- Cisco Service for IPS service contract

  Contact your reseller, Cisco service or product sales to purchase a contract. For more information, see Service Programs for IPS Products, page 4-40.

- Your IPS device serial number

  To find the IPS device serial number in IDM, click **Configuration > Licensing**, or in the CLI use the **show version** command.

- Valid Cisco.com username and password

Trial license keys are also available. If you cannot get your sensor licensed because of problems with your contract, you can obtain a 60-day trial license that supports signature updates that require licensing.

You can obtain a license key from the Cisco.com licensing server, which is then delivered to the sensor. Or, you can update the license key from a license key provided in a local file. Go to http://www.cisco.com/go/license and click **IPS Signature Subscription Service** to apply for a license key. For the procedure, see Obtaining and Installing the License Key, page 4-41.

You can view the status of the license key on the Licensing panel in IDM. Whenever you start IDM, you are informed of your license status—whether you have a trial, invalid, or expired license key. With no license key, an invalid license key, or an expired license key, you can continue to use IDM but you cannot download signature updates.

When you enter the CLI, you are informed of your license status. For example, you receive the following message if there is no license installed:

```
***LICENSE NOTICE***
There is no license key installed on the system.
The system will continue to operate with the currently installed
signature set. A valid license must be obtained in order to apply
signature updates. Please go to http://www.cisco.com/go/license
to obtain a new license or install a license.
```

You will continue to see this message until you install a license key.

Service Programs for IPS Products

You must have a Cisco Services for IPS service contract for any IPS product so that you can download a license key and obtain the IPS signature updates. If you have a direct relationship with Cisco Systems, contact your account manager or service account manager to purchase the Cisco Services for IPS service contract. If you do not have a direct relationship with Cisco Systems, you can purchase the service account from a one-tier or two-tier partner.

When you purchase the following IPS products you must also purchase a Cisco Services for IPS service contract:

- IDS-4215
- IPS-4240
- IPS-4255
- IPS-4260
- IDSM-2
- NM-CIDS

For ASA products, if you purchased one of the following ASA products that do not contain IPS, you must purchase a SMARTnet contract:

- ASA5510-K8
- ASA5510-DC-K8
- ASA5510-SEC-BUN-K9
- ASA5520-K8
- ASA5520-DC-K8
- ASA5520-BUN-K9
- ASA5540-K8
- ASA5540-DC-K8
- ASA5540-BUN-K9

Note SMARTnet provides operating system updates, access to Cisco.com, access to TAC, and hardware replacement NBD on site.

If you purchased one of the following ASA products that ships with the AIP-SSM installed or if you purchased AIP-SSM to add to your ASA product, you must purchase the Cisco Services for IPS service contract:

- ASA5510-AIP10-K9
- ASA5520-AIP10-K9
- ASA5520-AIP20-K9
- ASA5540-AIP20-K9
- ASA-SSM-AIP-10-K9

- ASA-SSM-AIP-20-K9

Note Cisco Services for IPS service contracts provide IPS signature updates, operating system updates, access to Cisco.com, access to TAC, and hardware replacement NBD on site.

For example, if you purchased an ASA-5510 and then later wanted to add IPS and purchased an ASA-SSM-AIP-10-K9, you must now purchase the Cisco Services for IPS service contract.

After you have the Cisco Services for IPS service contract, you must also have your product serial number to apply for the license key. For the procedure, see Obtaining and Installing the License Key, page 4-41.

Caution If you ever RMA your product, the serial number will change. You must then get a new license key for the new serial number.

Obtaining and Installing the License Key

Use the **copy** *source-url license_file_name* **license-key** command to copy the license key to your sensor.

The following options apply:

- *source-url*—The location of the source file to be copied. It can be a URL or keyword.
- *destination-url*—The location of the destination file to be copied. It can be a URL or a keyword.
- **license-key**—The subscription license file.
- *license_file_name*—The name of the license file you receive.

Note You cannot install an older license key over a newer license key.

The exact format of the source and destination URLs varies according to the file. Here are the valid types:

- ftp:—Source or destination URL for an FTP network server. The syntax for this prefix is:

  ftp:[//[username@] location]/relativeDirectory]/filename

  ftp:[//[username@]location]//absoluteDirectory]/filename
- scp:—Source or destination URL for the SCP network server. The syntax for this prefix is:

  scp:[//[username@] location]/relativeDirectory]/filename

  scp:[//[username@] location]//absoluteDirectory]/filename
- http:—Source URL for the web server. The syntax for this prefix is:

  http:[[/[username@]location]/directory]/filename
- https:—Source URL for the web server. The syntax for this prefix is:

  https:[[/[username@]location]/directory]/filename

Note If you use FTP or SCP, you are prompted for a password.

Note If you use SCP, the remote host must be on the SSH known hosts list. For the procedure, see Adding Hosts to the Known Hosts List, page 4-32.

Note If you use HTTPS, the remote host must be a TLS trusted host. For the procedure, see Adding TLS Trusted Hosts, page 4-37.

To install the license key, follow these steps:

Step 1 Apply for the license key at www.cisco.com/go/license.

Note In addition to a valid Cisco.com username and password, you must also have a Cisco Services for IPS service contract before you can apply for a license key. For more information, see Service Programs for IPS Products, page 4-40.

Step 2 Fill in the required fields.

Note You must have the correct IPS device serial number because the license key only functions on the device with that number.

Your Cisco IPS Signature Subscription Service license key is sent to the e-mail address you specified.

Step 3 Save the license key to a system that has a web server, FTP server, or SCP server.

Step 4 Log in to the CLI using an account with administrator privileges.

Step 5 Copy the license key to the sensor:

```
sensor# copy scp://user@10.89.147.3://tftpboot/dev.lic license-key
Password: *******
```

Step 6 Verify the sensor is licensed:

```
sensor# show version
Application Partition:

Cisco Intrusion Prevention System, Version 5.1(1)S182.0

OS Version 2.4.26-IDS-smp-bigphys
Platform: IPS-4255-K9
Serial Number: JAB0815R0JS
Licensed, expires: 19-Dec-2005 UTC
Sensor up-time is 2 days.
Using 706699264 out of 3974291456 bytes of available memory (17% usage)
system is using 17.3M out of 29.0M bytes of available disk space (59% usage)
application-data is using 36.5M out of 166.8M bytes of available disk space (23% usage)
boot is using 39.4M out of 68.6M bytes of available disk space (61% usage)


MainApp          2005_Feb_18_03.00   (Release)   2005-02-18T03:13:47-0600   Running
AnalysisEngine   2005_Feb_15_03.00   (QATest)    2005-02-15T12:59:35-0600   Running
CLI              2005_Feb_18_03.00   (Release)   2005-02-18T03:13:47-0600


Upgrade History:
```

```
    IDS-K9-maj-5.1-1-   14:16:00 UTC Thu Mar 04 2004

Recovery Partition Version 1.1 - 5.1(1)S182


sensor#
```

Step 7 Copy your license key from a sensor to a server to keep a backup copy of the license:

```
sensor# copy license-key scp://user@10.89.147.3://tftpboot/dev.lic
Password: *******
sensor#
```

CHAPTER 5

Configuring Interfaces

This chapter describes how to configure interfaces on the sensor. You configured the interfaces when you initialized the sensor with the **setup** command, but if you need to change or add anything to your interface configuration, use the following procedures. For the easiest way to configure interfaces, see Initializing the Sensor, page 3-2.

This chapter contains the following sections:

Understanding Interfaces

The sensor interfaces are named according to the maximum speed and physical location of the interface. The physical location consists of a port number and a slot number. All interfaces that are built-in on the sensor motherboard are in slot 0, and the PCI expansion slots are numbered beginning with slot 1 for the bottom slot with the slot numbers increasing from bottom to top. Interfaces with a given slot are numbered beginning with port 0 for the right port with the port numbers increasing from right to left. For example, GigabitEthernet2/1 supports a maximum speed of 1 Gigabit and is the second-from-the-right interface in the second-from-the bottom PCI expansion slot. IPS-4240 and IPS-4255 are exceptions to this rule. The command and control interface on these sensors is called Management0/0 rather than GigabitEthernet0/0.

There are three interface roles:

- Command and control
- Sensing
- Alternate TCP reset

There are restrictions on which roles you can assign to specific interfaces and some interfaces have multiple roles. You can configure any sensing interface to any other sensing interface as its TCP reset interface. The TCP reset interface can also serve as an IDS (promiscuous) sensing interface at the same time. The following restrictions apply:

- Because NM-CIDS and AIP-SSM only have one sensing interface, you cannot configure a TCP reset interface.
- Because of hardware limitations on the Catalyst switch, both of the IDSM-2 sensing interfaces are permanently configured to use System0/1 as the TCP reset interface.
- The TCP reset interface that is assigned to a sensing interface has no effect in inline interface or inline VLAN pair mode, because TCP resets are always sent on the sensing interfaces in those modes.

This section contains the following topics:

Command and Control Interface

The command and control interface has an IP address and is used for configuring the sensor. It receives security and status events from the sensor and queries the sensor for statistics.

The command and control interface is permanently enabled. It is permanently mapped to a specific physical interface, which depends on the specific model of sensor. You cannot use the command and control interface as either a sensing or alternate TCP reset interface.

Table 5-1 lists the command and control interfaces for each sensor.

Table 5-1 Command and Control Interfaces

Sensor	Command and Control Interface
IDS-4210	FastEthernet0/1
IDS-4215	FastEthernet0/0
IDS-4235	GigabitEthernet0/1
IDS-4250	GigabitEthernet0/1
IPS-4240	Management0/0
IPS-4255	Management0/0
IPS-4260	Management0/0
NM-CIDS	FastEthernet0/0
AIP-SSM-10	GigabitEthernet0/0

Table 5-1 Command and Control Interfaces (continued)

Sensor	Command and Control Interface
AIP-SSM-20	GigabitEthernet0/0
IDSM-2	GigabitEthernet0/2

Sensing Interfaces

Sensing interfaces are used by the sensor to analyze traffic for security violations. A sensor has one or more sensing interfaces depending on the sensor. For the number and type of sensing interfaces available for each sensor, see Interface Support, page 5-3.

Sensing interfaces can operate individually in promiscuous mode or you can pair them to create inline interfaces for inline sensing mode. For more information, see Understanding Promiscuous Mode, page 5-14, Understanding Inline Interface Mode, page 5-15, and Understanding Inline VLAN Pair Mode, page 5-17.

Note On appliances, all sensing interfaces are disabled by default. You must enable them to use them. On modules, the sensing interfaces are permanently enabled.

Some appliances support optional PCI interface cards that add sensing interfaces to the sensor. You must insert or remove these optional cards while the sensor is powered off. The sensor detects the addition or removal of a supported interface card. If you remove an optional PCI card, some of the interface configuration is deleted, such as the speed, duplex, description string, enabled/disabled state of the interface, and any inline interface pairings. These settings are restored to their default settings when the card is reinstalled. However, the assignment of promiscuous and inline interfaces to the Analysis Engine is not deleted from the Analysis Engine configuration, but is ignored until those cards are reinserted and you create the inline interface pairs again. For more information, see Assigning Interfaces to the Virtual Sensor, page 5-24.

Interface Support

Table 5-2 describes the interface support for appliances and modules running IPS 5.1:

Table 5-2 Interface Support

Base Chassis	Added PCI Cards	Interfaces Supporting Inline VLAN Pairs (Sensing Ports)	Combinations Supporting Inline Interface Pairs	Interfaces Not Supporting Inline (Command and Control Port)
IDS-4210	—	FastEthernet0/0	N/A	FastEthernet0/1
IDS-4215	—	FastEthernet0/1	N/A	FastEthernet0/0

Table 5-2 Interface Support (continued)

Base Chassis	Added PCI Cards	Interfaces Supporting Inline VLAN Pairs (Sensing Ports)	Combinations Supporting Inline Interface Pairs	Interfaces Not Supporting Inline (Command and Control Port)
IDS-4215	4FE	FastEthernet0/1 FastEthernetS/0[1] FastEthernetS/1 FastEthernetS/2 FastEthernetS/3	1/0<->1/1 1/0<->1/2 1/0<->1/3 1/1<->1/2 1/1<->1/3 1/2<->1/3 0/1<->1/0 0/1<->1/1 0/1<->1/2 0/1<->1/3	FastEthernet0/0
IDS-4235	—	GigabitEthernet0/0	N/A	GigabitEthernet0/1
IDS-4235	4FE	GigabitEthernet0/0 FastEthernetS/0 FastEthernetS/1 FastEthernetS/2 FastEthernetS/3	1/0<->1/1 1/0<->1/2 1/0<->1/3 1/1<->1/2 1/1<->1/3 1/2<->1/3	GigabitEthernet0/1
IDS-4235	TX (GE)	GigabitEthernet0/0 GigabitEthernet1/0 GigabitEthernet2/0	0/0<->1/0 0/0<->2/0	GigabitEthernet0/1
IDS-4250	—	GigabitEthernet0/0	N/A	GigabitEthernet0/1
IDS-4250	4FE	GigabitEthernet0/0 FastEthernetS/0 FastEthernetS/1 FastEthernetS/2 FastEthernetS/3	1/0<->1/1 1/0<->1/2 1/0<->1/3 1/1<->1/2 1/1<->1/3 1/2<->1/3	GigabitEthernet0/1
IDS-4250	TX (GE)	GigabitEthernet0/0 GigabitEthernet1/0 GigabitEthernet2/0	0/0<->1/0 0/0<->2/0	GigabitEthernet0/1
IDS-4250	SX	GigabitEthernet0/0 GigabitEthernet1/0	N/A	GigabitEthernet0/1
IDS-4250	SX + SX	GigabitEthernet0/0 GigabitEthernet1/0 GigabitEthernet2/0	1/0<->2/0	GigabitEthernet0/1
IDS-4250	XL	GigabitEthernet0/0 GigabitEthernet2/0 GigabitEthernet2/1	2/0<->2/1	GigabitEthernet0/1
IDSM-2	—	GigabitEthernet0/7 GigabitEthernet0/8	0/7<->0/8	GigabitEthernet0/2

Table 5-2 Interface Support (continued)

Base Chassis	Added PCI Cards	Interfaces Supporting Inline VLAN Pairs (Sensing Ports)	Combinations Supporting Inline Interface Pairs	Interfaces Not Supporting Inline (Command and Control Port)
IPS-4240	—	GigabitEthernet0/0 GigabitEthernet0/1 GigabitEthernet0/2 GigabitEthernet0/3	0/0<->0/1 0/0<->0/2 0/0<->0/3 0/1<->0/2 0/1<->0/3 0/2<->0/3	Management0/0
IPS-4255	—	GigabitEthernet0/0 GigabitEthernet0/1 GigabitEthernet0/2 GigabitEthernet0/3	0/0<->0/1 0/0<->0/2 0/0<->0/3 0/1<->0/2 0/1<->0/3 0/2<->0/3	Management0/0
IPS-4260	—	GigabitEthernet0/1	N/A	Management0/0
IPS-4260	4GE-BP Slot 1 Slot 2	GigabitEthernet0/1 GigabitEthernet2/0 GigabitEthernet2/1 GigabitEthernet2/2 GigabitEthernet2/3 GigabitEthernet3/0 GigabitEthernet3/1 GigabitEthernet3/2 GigabitEthernet3/3	 2/0<->2/1[2] 2/2<->2/3 3/0<->3/1 3/2<->3/3	Management0/0
IPS-4260	2SX Slot 1 Slot 2	GigabitEthernet0/1 GigabitEthernet2/0 GigabitEthernet2/1 GigabitEthernet2/2 GigabitEthernet2/3 GigabitEthernet3/0 GigabitEthernet3/1 GigabitEthernet3/2 GigabitEthernet3/3	All sensing ports can be paired together	Management0/0
NM-CIDS	—	None	N/A	All

Table 5-2 Interface Support (continued)

Base Chassis	Added PCI Cards	Interfaces Supporting Inline VLAN Pairs (Sensing Ports)	Combinations Supporting Inline Interface Pairs	Interfaces Not Supporting Inline (Command and Control Port)
AIP-SSM-10	—	GigabitEthernet0/1 by security context instead of VLAN pair or inline interface pair	GigabitEthernet0/1 by security context instead of VLAN pair or inline interface pair	GigabitEthernet0/0
AIP-SSM-20	—	GigabitEthernet0/1 by security context instead of VLAN pair or inline interface pair	GigabitEthernet0/1 by security context instead of VLAN pair or inline interface pair	GigabitEthernet0/0

1. You can install the 4FE card in either slot 1 or 2. S indicates the slot number, which can be either 1 or 2.
2. You can pair any two sensing ports together if you are not using the hardware bypass feature.

TCP Reset Interfaces

This section explains the TCP reset interfaces and when to use them. It contains the following topics:

- Understanding Alternate TCP Reset Interfaces, page 5-6
- Designating the Alternate TCP Reset Interface, page 5-7

Understanding Alternate TCP Reset Interfaces

You can configure sensors to send TCP reset packets to try to reset a network connection between an attacker host and its intended target host. In some installations when the interface is operating in promiscuous mode, the sensor may not be able to send the TCP reset packets over the same sensing interface on which the attack was detected. In such cases, you can associate the sensing interface with an alternate TCP reset interface and any TCP resets that would otherwise be sent on the sensing interface when it is operating in promiscuous mode, are instead sent out on the associated alternate TCP reset interface. For more information, see Designating the Alternate TCP Reset Interface, page 5-7.

If a sensing interface is associated with an alternate TCP reset interface, that association applies when the sensor is configured for promiscuous mode but is ignored when the sensing interface is configured for inline mode.

With the exception of IDSM-2, any sensing interface can serve as the alternate TCP reset interface for another sensing interface. The alternate TCP reset interface on IDSM-2 is fixed because of hardware limitation.

Table 5-3 lists the alternate TCP reset interfaces.

Table 5-3 Alternate TCP Reset Interfaces

Sensor	Alternate TCP Reset Interface
IDS-4210	None[1]
IDS-4215	Any sensing interface

Table 5-3 Alternate TCP Reset Interfaces

Sensor	Alternate TCP Reset Interface
IDS-4235	Any sensing interface
IDS-4250	Any sensing interface
IPS-4240	Any sensing interface
IPS-4255	Any sensing interface
IPS-4260	Any sensing interface
NM-CIDS	None[2]
AIP-SSM-10	None[3]
AIP-SSM-20	None[4]
IDSM-2	System0/1[5]

1. There is only one sensing interface on IDS-4210.
2. There is only one sensing interface on NM-CIDS.
3. There is only one sensing interface on AIP-SSM-10.
4. There is only one sensing interface on AIP-SSM-20.
5. This is an internal interface on the Catalyst backplane.

Designating the Alternate TCP Reset Interface

You need to designate an alternate TCP reset interface in the following situations:

- When a switch is being monitored with either SPAN or VACL capture and the switch does not accept incoming packets on the SPAN or VACL capture port.
- When a switch is being monitored with either SPAN or VACL capture for multiple VLANs, and the switch does not accept incoming packets with 802.1q headers.

  Note The TCP resets need 802.1q headers to tell which VLAN the resets should be sent on.

- When a network tap is used for monitoring a connection.

  Note Taps do not permit incoming traffic from the sensor.

  You can only assign a sensing interface as an alternate TCP reset interface. You cannot configure the management interface as an alternate TCP reset interface.

Hardware Bypass Mode

In addition to IPS 5.1 software bypass, IPS-4260 also supports hardware bypass.

This section describes the hardware bypass card and its configuration restrictions. For the procedure for installing and removing the hardware bypass card, refer to Installing and Removing PCI Cards.

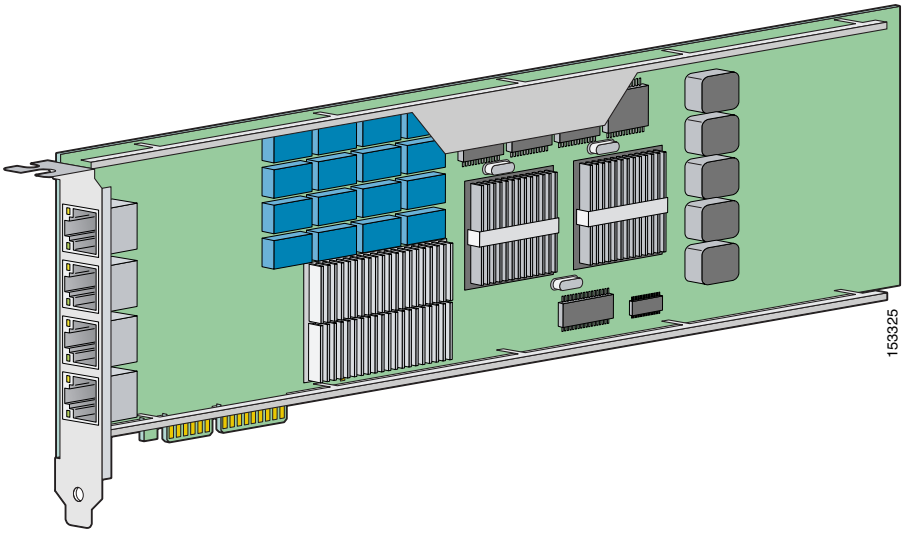
This section contains the following topics:

Hardware Bypass Card

IPS-4260 supports the Intel 4-port PCI-Express card (part number IPS-4GE-BP-INT=) with hardware bypass. This 4GE bypass interface card supports hardware bypass only between ports 0 and 1 and between ports 2 and 3.

Figure 5-1 shows the 4GE bypass interface card.

Figure 5-1 4GE Bypass Interface Card



Hardware bypass complements the existing software bypass feature in IPS 5.1. For more information on software bypass mode, see Inline Bypass Mode, page 5-22. The following conditions apply to hardware bypass and software bypass on IPS-4260:

- When bypass is set to OFF, software bypass is not active.

  For each inline interface for which hardware bypass is available, the component interfaces are set to disable the fail-open capability. If SensorApp fails, the sensor is powered off, reset, or if the NIC interface drivers fail or are unloaded, the paired interfaces enter the fail-closed state (no traffic flows through inline interface or inline VLAN subinterfaces).

- When bypass is set to ON, software bypass is active.

  Software bypass forwards packets between the paired physical interfaces in each inline interface and between the paired VLANs in each inline VLAN subinterface. For each inline interface on which hardware bypass is available, the component interfaces are set to standby mode. If the sensor is powered off, reset, or if the NIC interfaces fail or are unloaded, those paired interfaces enter fail-open state in hardware. Any other inline interfaces enter fail-closed state.

- When bypass is set to AUTO (traffic flows without inspection), software bypass is activated if sensorApp fails.

  For each inline interface on which hardware bypass is available, the component interfaces are set to standby mode. If the sensor is powered off, reset, or if the NIC interfaces fail or are unloaded, those paired interfaces enter fail-open state in hardware. Any other inline interfaces enter the fail-closed state.

Note To test fail-over, set the bypass mode to ON or AUTO, create one or more inline interfaces and power down the sensor and verify that traffic still flows through the inline path.

Hardware Bypass Configuration Restrictions

To use the hardware bypass feature on the 4GE bypass interface card, you must pair interfaces to support the hardware design of the card. If you create an inline interface that pairs a hardware-bypass-capable interface with an interface that violates one or more of the hardware-bypass configuration restrictions, hardware bypass is deactivated on the inline interface and you receive a warning message similar to the following:

```
Hardware bypass functionality is not available on Inline-interface pair0.
Physical-interface GigabitEthernet1/0 is capable of performing hardware bypass only when
paired with GigabitEthernet1/1, and both interfaces are enabled and configured with the
same speed and duplex settings.
```

The following configuration restrictions apply to hardware bypass:

- The 4-port bypass card is only supported on IPS-4260.
- Fail-open hardware bypass only works on inline interfaces (interface pairs), not on inline VLAN pairs.
- Fail-open hardware bypass is available on an inline interface if all of the following conditions are met:
  - Both of the physical interfaces support hardware bypass.
  - Both of the physical interfaces are on the same interface card.
  - The two physical interfaces are associated in hardware as a bypass pair.
  - The speed and duplex settings are identical on the physical interfaces.
  - Both of the interfaces are administratively enabled.

Configuration Sequence

Follow these steps to configure interfaces on the sensor:

1. Configure the physical interface settings (speed, duplex, and admin state).

   For the procedure, see Configuring Physical Interfaces, page 5-11.

2. Create or delete inline interfaces and/or inline VLAN subinterfaces, and set the inline bypass mode.

   For more information, see Inline Interface Mode, page 5-15, Inline VLAN Pair Mode, page 5-17, and Inline Bypass Mode, page 5-22.

3. Assign the physical, subinterfaces, and inline interfaces to the virtual sensor.

   For the procedure, see Assigning Interfaces to the Virtual Sensor, page 5-24.

Interface Configuration Restrictions

For hardware bypass interface configuration restrictions, see Hardware Bypass Mode, page 5-7.

The following restrictions apply to configuring interfaces on the sensor:

- Physical Interfaces
  - On modules (IDSM-2, NM-CIDS, AIP-SSM-10, and AIP-SSM-20) and IPS-4240, IPS-4255, and IPS-4260, all backplane interfaces have fixed speed, duplex, and state settings. These settings are protected in the default configuration on all backplane interfaces.
  - For nonbackplane FastEthernet interfaces the valid speed settings are 10 Mbps, 100 Mbps, and auto. Valid duplex settings are full, half, and auto.
  - For Gigabit fiber interfaces (1000-SX and XL on the IDS-4250), valid speed settings are 1000 Mbps and auto.
  - For Gigabit copper interfaces (1000-TX on the IDS-4235, IDS-4250, IPS-4240, IPS-4255, and IPS-4260), valid speed settings are 10 Mbps, 100 Mbps, 1000 Mbps, and auto. Valid duplex settings are full, half, and auto.
  - For Gigabit (copper or fiber) interfaces, if the speed is configured for 1000 Mbps, the only valid duplex setting is auto.
  - The command and control interface cannot also serve as a sensing interface.
- Inline Interface Pairs
  - Inline interface pairs can contain any combination of sensing interfaces regardless of the physical interface type (copper versus fiber), speed, or duplex settings of the interface. However, pairing interfaces of different media type, speeds, and duplex settings may not be fully tested or supported. For more information, see Interface Support, page 5-3.
  - The command and control interface cannot be a member of an inline interface pair.
  - You cannot pair a physical interface with itself in an inline interface pair.
  - A physical interface can be a member of only one inline interface pair.
  - You can only configure bypass mode and create inline interface pairs on sensor platforms that support inline mode.
  - A physical interface cannot be a member of an inline interface pair unless the subinterface mode of the physical interface is **none**.
- Inline VLAN Interface Pairs
  - You cannot pair a VLAN with itself.
  - For a given sensing interface, a VLAN can be a member of only one inline VLAN pair. However, a given VLAN can be a member of an inline VLAN pair on more than one sensing interface.
  - The order in which you specify the VLANs in an inline VLAN pair is not significant.
  - A sensing interface in inline VLAN pair mode can have from 1 to 255 inline VLAN pairs.
- Alternate TCP Reset Interface:
  - You can only assign the alternate TCP reset interface to a sensing interface. You cannot configure the command and control interface as an alternate TCP reset interface. The alternate TCP reset interface option is set to **none** as the default and is protected for all interfaces except the sensing interfaces.

- You can assign the same physical interface as an alternate TCP reset interface for multiple sensing interfaces.
- A physical interface can serve as both a sensing interface and an alternate TCP reset interface.
- The command and control interface cannot serve as the alternate TCP reset interface for a sensing interface.
- A sensing interface cannot serve as its own alternate TCP reset interface.
- You can only configure interfaces that are capable of TCP resets as alternate TCP reset interfaces.

Note The exception to this restriction is the IDSM-2. The alternate TCP reset interface assignments for both sensing interfaces is System0/1 (protected).

Configuring Physical Interfaces

Note For information on what you need to configure if you are using the hardware bypass card on IPS-4260, see Hardware Bypass Configuration Restrictions, page 5-9.

Use the **physical-interfaces** *interface_name* command in the service interface submode to configure promiscuous interfaces. The interface name is FastEthernet or GigabitEthernet. For a list of possible interfaces for your sensor, see Interface Support, page 5-3.

Note AIP-SSM is configured for promiscuous mode from the ASA CLI and not from the IPS CLI. For the procedure, see Sending Traffic to AIP-SSM, page 14-2.

The following options apply:

- **admin-state {enabled | disabled}**—The administrative link state of the interface, whether the interface is enabled or disabled.

  Note On all backplane sensing interfaces on all modules (IDSM-2 NM-CIDS, and AIP-SSM), **admin-state** is set to enabled and is protected (you cannot change the setting). The **admin-state** has no effect (and is protected) on the command and control interface. It only affects sensing interfaces. The command and control interface does not need to be enabled because it cannot be monitored.

- **alt-tcp-reset-interface**—Sends TCP resets out an alternate interface when this interface is used for promiscuous monitoring and the reset action is triggered by a signature firing. For more information on the alternate TCP reset interface, see Understanding Alternate TCP Reset Interfaces, page 5-6, and Designating the Alternate TCP Reset Interface, page 5-7.

  Note You can only assign a sensing interface as an alternate TCP reset interface. You cannot configure the management interface as an alternate TCP reset interface.

Note This option is protected on modules (IDSM-2 NM-CIDS, and AIP-SSM) and appliances that only have one sensing interface (IDS-4210, IDS-4215,IDS-4235, and IDS-4250 without any additional NIC cards).

  - *interface_name*—The name of the interface on which TCP resets should be sent when this interface is used for promiscuous monitoring and the reset action is triggered by a signature firing. This setting is ignored when this interface is a member of an inline interface.
  - **none** —Disables the use of an alternate TCP reset interface. TCP resets triggered by the reset action when in promiscuous mode will be sent out of this interface instead.
- **default**—Sets the value back to the system default setting.
- **description**—Your description of the promiscuous interface.
- **duplex**—The duplex setting of the interface.
  - **auto**—Sets the interface to auto negotiate duplex.
  - **full**—Sets the interface to full duplex.
  - **half**—Sets the interface to half duplex.

Note The **duplex** option is protected on all modules.

- **no**—Remove an entry or selection setting.
- **speed**—The speed setting of the interface.
  - **auto**—Sets the interface to auto negotiate speed.
  - **10**—Sets the interface to 10 MB (for TX interfaces only).
  - **100**—Sets the interface to 100 MB (for TX interfaces only).
  - **1000**—Sets the interface to 1 GB (for Gigabit interfaces only).

Note The **speed** option is protected on all modules.

To configure the promiscuous interface settings on the sensor, follow these steps:

Step 1 Log in to the CLI using an account with administrator privileges.

Step 2 Enter interface submode:

```
sensor# configure terminal
sensor(config)# service interface
```

Step 3 Display the list of available interfaces:

```
sensor(config-int)# physical-interfaces ?
GigabitEthernet0/0     GigabitEthernet0/0 physical interface.
GigabitEthernet0/1     GigabitEthernet0/1 physical interface.
GigabitEthernet0/2     GigabitEthernet0/2 physical interface.
GigabitEthernet0/3     GigabitEthernet0/3 physical interface.
Management0/0          Management0/0 physical interface.
sensor(config-int)# physical-interfaces
```

Step 4 Specify the interface for promiscuous mode:

```
sensor(config-int)# physical-interfaces GigabitEthernet0/2
```

Step 5 Enable the interface:

```
sensor(config-int-phy)# admin-state enabled
```

The interface must be assigned to the virtual sensor (see Assigning Interfaces to the Virtual Sensor, page 5-24) and enabled to monitor traffic.

Step 6 Add a description of this interface:

```
sensor(config-int-phy)# description INT1
```

Step 7 Configure the duplex settings:

```
sensor(config-int-phy)# duplex full
```

This option is not available on modules.

Step 8 Configure the speed:

```
sensor(config-int-phy)# speed 1000
```

This option is not available on modules.

Step 9 Enable TCP resets for this interface if desired:

```
sensor(config-int-phy)# alt-tcp-reset-interface interface-name GigabitEthernet2/0
```

Step 10 Repeat Steps 4 through 9 for any other interfaces you want to designate as promiscuous interfaces.

Step 11 Verify the settings:

Note Make sure the `subinterface-type` is *none*, the default. You use the **subinterface-type** command to configure inline VLAN pairs. For more information, see Configuring Inline VLAN Pairs, page 5-18.

```
sensor(config-int-phy)# show settings
   <protected entry>
   name: GigabitEthernet0/2
   ----------------------------------------------
      media-type: tx <protected>
      description: INT1 default:
      admin-state: enabled default: disabled
      duplex: full default: auto
      speed: 1000 default: auto
     alt-tcp-reset-interface
     ---------------------------------------------
        interface-name: GigabitEthernet2/0
     ---------------------------------------------
     subinterface-type
     ---------------------------------------------
        none
        ---------------------------------------------
        ---------------------------------------------
     ---------------------------------------------
  ----------------------------------------------
sensor(config-int-phy)#
```

Step 12 To remove TCP resets from an interface:

```
sensor(config-int-phy)# alt-tcp-reset-interface none
```

Step 13 Verify the settings:

```
sensor(config-int-phy)# show settings
   <protected entry>
   name: GigabitEthernet0/0
   ----------------------------------------------
      media-type: tx <protected>
      description: <defaulted>
      admin-state: disabled <protected>
      duplex: auto <defaulted>
      speed: auto <defaulted>
      alt-tcp-reset-interface
      ----------------------------------------------
         none
         ----------------------------------------------
         ----------------------------------------------
      ----------------------------------------------
   ----------------------------------------------
sensor(config-int-phy)#
```

Step 14 Exit interface submode:

```
sensor(config-int-phy)# exit
sensor(config-int)# exit
Apply Changes:?[yes]:
```

Step 15 Press **Enter** to apply the changes or enter **no** to discard them.

Promiscuous Mode

This section describes promiscuous mode on the sensor, and contains the following topics:

- Understanding Promiscuous Mode, page 5-14
- Configuring Promiscuous Mode, page 5-15

Understanding Promiscuous Mode

In promiscuous mode, packets do not flow through the sensor. The sensor analyzes a copy of the monitored traffic rather than the actual forwarded packet. The advantage of operating in promiscuous mode is that the sensor does not affect the packet flow with the forwarded traffic. The disadvantage of operating in promiscuous mode, however, is the sensor cannot stop malicious traffic from reaching its intended target for certain types of attacks, such as atomic attacks (single-packet attacks). The response actions implemented by promiscuous sensor devices are post-event responses and often require assistance from other networking devices, for example, routers and firewalls, to respond to an attack. While such response actions can prevent some classes of attacks, in atomic attacks the single packet has the chance of reaching the target system before the promiscuous-based sensor can apply an ACL modification on a managed device (such as a firewall, switch, or router).

Configuring Promiscuous Mode

By default, all sensing interfaces are in promiscuous mode. To change an interface from inline mode to promiscuous mode, delete any inline interface that contains that interface and delete any inline VLAN pair subinterfaces of that interface from the interface configuration.

Inline Interface Mode

This section describes inline mode on the sensor, and contains the following topics:

- Understanding Inline Interface Mode, page 5-15
- Configuring Inline Interface Pairs, page 5-15

Understanding Inline Interface Mode

Operating in inline interface mode puts the IPS directly into the traffic flow and affects packet-forwarding rates making them slower by adding latency. This allows the sensor to stop attacks by dropping malicious traffic before it reaches the intended target, thus providing a protective service. Not only is the inline device processing information on layers 3 and 4, but it is also analyzing the contents and payload of the packets for more sophisticated embedded attacks (layers 3 to 7). This deeper analysis lets the system identify and stop and/or block attacks that would normally pass through a traditional firewall device.

In inline interface mode, a packet comes in through the first interface of the pair on the sensor and out the second interface of the pair. The packet is sent to the second interface of the pair unless that packet is being denied or modified by a signature.

Note You can configure AIP-SSM to operate inline even though it has only one sensing interface.

Note If the paired interfaces are connected to the same switch, you should configure them on the switch as access ports with different access VLANs for the two ports. Otherwise, traffic does not flow through the inline interface.

Configuring Inline Interface Pairs

Note For information on what you need to configure if you are using the hardware bypass card on IPS-4260, see Hardware Bypass Configuration Restrictions, page 5-9.

Use the **inline-interfaces** command in the service interface submode to configure inline interfaces.

Note AIP-SSM is configured for inline interface mode from the ASA CLI and not from the IPS CLI. For the procedure, see Sending Traffic to AIP-SSM, page 14-2.

The following options apply:

- **inline-interfaces**—Name of the logical inline interface pair.
- **default**—Sets the value back to the system default setting.
- **description**—Your description of the inline interface pair.
- **interface1**—The first interface in the inline interface pair.
- **interface2**—The second interface in the inline interface pair.
- **no**—Removes an entry or selection setting.

To configure the inline interface settings, follow these steps:

Step 1 Log in to the CLI using an account with administrator privileges.

Step 2 Enter interface submode:

```
sensor# configure terminal
sensor(config)# service interface
sensor(config-int)#
```

Step 3 Verify that the subinterface mode is "none" for both of the physical interfaces you are pairing in the inline interface:

```
sensor(config-int)# show settings
   physical-interfaces (min: 0, max: 999999999, current: 2)
   ----------------------------------------------
      <protected entry>
      name: GigabitEthernet0/0 <defaulted>
      ---------------------------------------------
         media-type: tx <protected>
         description: <defaulted>
         admin-state: disabled <protected>
         duplex: auto <defaulted>
         speed: auto <defaulted>
         alt-tcp-reset-interface
         ----------------------------------------------
            none
            ----------------------------------------------
            ----------------------------------------------
         ----------------------------------------------
         subinterface-type
         ----------------------------------------------
            none
            ----------------------------------------------
            ----------------------------------------------
         ----------------------------------------------
      ---------------------------------------------
```

Step 4 Name the inline pair:

```
sensor(config-int)# inline-interfaces PAIR1
```

Step 5 Display the available interfaces:

```
sensor(config-int)# interface1?
GigabitEthernet0/0     GigabitEthernet0/0 physical interface.
GigabitEthernet0/1     GigabitEthernet0/1 physical interface.
GigabitEthernet0/2     GigabitEthernet0/2 physical interface.
GigabitEthernet0/3     GigabitEthernet0/3 physical interface.
Management0/0          Management0/0 physical interface.
```

Step 6 Configure two interfaces into a pair:

```
sensor(config-int-inl)# interface1 GigabitEthernet0/0
sensor(config-int-inl)# interface2 GigabitEthernet0/1
```

Step 7 Add a description of the interface pair:

```
sensor(config-int-inl)# description pairs interfaces Gig0/0 and Gig0/1
```

Step 8 Repeat Steps 4 through 7 for any other interfaces that you want to configure into inline interface pairs.

Step 9 Verify the settings:

```
sensor(config-int-inl)# show settings
   name: PAIR1
   -----------------------------------------------
      description: PAIR1 = Gig0/0 & Gig0/1 default:
      interface1: GigabitEthernet0/0
      interface2: GigabitEthernet0/1
   -----------------------------------------------
```

Step 10 To remove an inline interface pair and return the interfaces to promiscuous mode:

```
sensor(config-int-inl)# exit
sensor(config-int)#
```

Step 11 Enable the interfaces assigned to the interface pair:

```
sensor(config-int)# physical-interfaces GigabitEthernet0/0
sensor(config-int-phy)# admin-state enabled
sensor(config-int-phy)# exit
sensor(config-int)# physical-interfaces GigabitEthernet0/1
sensor(config-int-phy)# admin-state enabled
sensor(config-int-phy)# exit
sensor(config-int)#
```

Step 12 Exit interface configuration submode:

```
sensor(config-int)# exit
Apply Changes:?[yes]:
```

Step 13 Press **Enter** to apply the changes or enter **no** to discard them.

Inline VLAN Pair Mode

This section describes inline VLAN pair mode and how to configure inline VLAN pairs. It contains the following topics:

- Understanding Inline VLAN Pair Mode, page 5-17
- Configuring Inline VLAN Pairs, page 5-18

Understanding Inline VLAN Pair Mode

Note For IPS-4260, fail-open hardware bypass is not supported on inline VLAN pairs. For more information, see Hardware Bypass Configuration Restrictions, page 5-9.

You can associate VLANs in pairs on a physical interface. Packets received on one of the paired VLANs are analyzed and then forwarded to the other VLAN in the pair. Inline VLAN pairs are supported on all sensors that are compatible with IPS 5.1 except NM-CIDS, AIP-SSM-10, and AIP-SSM-20.

Inline VLAN pair mode is an active sensing mode where a sensing interface acts as an 802.1q trunk port, and the sensor performs VLAN bridging between pairs of VLANs on the trunk. The sensor inspects the traffic it receives on each VLAN in each pair, and can either forward the packets on the other VLAN in the pair, or drop the packet if an intrusion attempt is detected. You can configure an IPS sensor to simultaneously bridge up to 255 VLAN pairs on each sensing interface. The sensor replaces the VLAN ID field in the 802.1q header of each received packet with the ID of the egress VLAN on which the sensor forwards the packet. The sensor drops all packets received on any VLANs that are not assigned to inline VLAN pairs.

Configuring Inline VLAN Pairs

Use the **physical-interfaces** *interface_name* command in the service interface submode to configure inline VLAN pairs. The interface name is FastEthernet or GigabitEthernet.

The following options apply:

- **admin-state {enabled | disabled}**—The administrative link state of the interface, whether the interface is enabled or disabled.

  Note On all backplane sensing interfaces on all modules (IDSM-2 NM-CIDS, and AIP-SSM), **admin-state** is set to enabled and is protected (you cannot change the setting). The **admin-state** has no effect (and is protected) on the command and control interface. It only affects sensing interfaces. The command and control interface does not need to be enabled because it cannot be monitored.

- **default**—Sets the value back to the system default setting.
- **description**—Your description of the interface.
- **duplex**—The duplex setting of the interface.
  - **auto**—Sets the interface to auto negotiate duplex.
  - **full**—Sets the interface to full duplex.
  - **half**—Sets the interface to half duplex.

    Note The **duplex** option is protected on all modules.

- **no**—Removes an entry or selection setting.

- **speed**—The speed setting of the interface.
  - **auto**—Sets the interface to auto negotiate speed.
  - **10**—Sets the interface to 10 MB (for TX interfaces only).
  - **100**—Sets the interface to 100 MB (for TX interfaces only).
  - **1000**—Sets the interface to 1 GB (for Gigabit interfaces only).

  Note The **speed** option is protected on all modules.

- **subinterface-type**—Specifies that the interface is a subinterface and what type of subinterface is defined.
  - **inline-vlan-pair**—Lets you define the subinterface as an inline VLAN pair.
  - **none**—No subinterfaces defined.
- **subinterface**—Defines the subinterface as an inline VLAN pair.
  - **vlan1**—The first VLAN in the inline VLAN pair.
  - **vlan2**—The second VLAN in the inline VLAN pair.

To configure the inline VLAN pair settings on the sensor, follow these steps:

Step 1 Log in to the CLI using an account with administrator privileges.

Step 2 Enter interface submode:

```
sensor# configure terminal
sensor(config)# service interface
sensor(config-int)#
```

Step 3 Verify if any inline interfaces exist (the subinterface type should read “none” if no inline interfaces have been configured):

```
sensor(config-int)# show settings
   physical-interfaces (min: 0, max: 999999999, current: 5)
   ------------------------------------------------
      <protected entry>
      name: GigabitEthernet0/0 <defaulted>
      ------------------------------------------------
         media-type: tx <protected>
         description: <defaulted>
         admin-state: disabled <defaulted>
         duplex: auto <defaulted>
         speed: auto <defaulted>
         alt-tcp-reset-interface
         ------------------------------------------------
            none
            ------------------------------------------------
            ------------------------------------------------
         ------------------------------------------------
         subinterface-type
         ------------------------------------------------
            none
            ------------------------------------------------
            ------------------------------------------------
         ------------------------------------------------
      ------------------------------------------------
      <protected entry>
      name: GigabitEthernet0/1 <defaulted>
```

```
-----------------------------------------------
   media-type: tx <protected>
   description: <defaulted>
   admin-state: disabled <defaulted>
   duplex: auto <defaulted>
   speed: auto <defaulted>
   alt-tcp-reset-interface
   ----------------------------------------------
      none
      ---------------------------------------------
      ---------------------------------------------
   ----------------------------------------------
   subinterface-type
   ----------------------------------------------
      none
      ---------------------------------------------
      ---------------------------------------------
   ----------------------------------------------
-----------------------------------------------
<protected entry>
name: GigabitEthernet0/2 <defaulted>
-----------------------------------------------
   media-type: tx <protected>
   description: <defaulted>
   admin-state: disabled <defaulted>
   duplex: auto <defaulted>
   speed: auto <defaulted>
   alt-tcp-reset-interface
   ----------------------------------------------
      none
      ---------------------------------------------
      ---------------------------------------------
   ----------------------------------------------
   subinterface-type
   ----------------------------------------------
      none
      ---------------------------------------------
      ---------------------------------------------
   ----------------------------------------------
-----------------------------------------------
<protected entry>
name: GigabitEthernet0/3 <defaulted>
-----------------------------------------------
   media-type: tx <protected>
   description: <defaulted>
   admin-state: disabled <defaulted>
   duplex: auto <defaulted>
   speed: auto <defaulted>
   alt-tcp-reset-interface
   ----------------------------------------------
      none
      ---------------------------------------------
      ---------------------------------------------
   ----------------------------------------------
   subinterface-type
   ----------------------------------------------
      none
      ---------------------------------------------
      ---------------------------------------------
   ----------------------------------------------
-----------------------------------------------
<protected entry>
name: Management0/0 <defaulted>
-----------------------------------------------
```

```
            media-type: tx <protected>
            description: <defaulted>
            admin-state: disabled <protected>
            duplex: auto <defaulted>
            speed: auto <defaulted>
            alt-tcp-reset-interface
            ----------------------------------------------
               none
               ---------------------------------------------
               ---------------------------------------------
            ----------------------------------------------
            subinterface-type
            ----------------------------------------------
               none
               ---------------------------------------------
               ---------------------------------------------
            ----------------------------------------------
         ---------------------------------------------
      ---------------------------------------------
      command-control: Management0/0 <protected>
      inline-interfaces (min: 0, max: 999999999, current: 0)
      ---------------------------------------------
      ---------------------------------------------
      bypass-mode: auto <defaulted>
      interface-notifications
      ---------------------------------------------
         missed-percentage-threshold: 0 percent <defaulted>
         notification-interval: 30 seconds <defaulted>
         idle-interface-delay: 30 seconds <defaulted>
      ---------------------------------------------
sensor(config-int)#
```

Step 4 If there are inline interfaces that are using this physical interface, remove them:

```
sensor(config-int)# no inline-interfaces interface_name
```

Step 5 Display the list of available interfaces:

```
sensor(config-int)# physical-interfaces ?
GigabitEthernet0/0     GigabitEthernet0/0 physical interface.
GigabitEthernet0/1     GigabitEthernet0/1 physical interface.
GigabitEthernet0/2     GigabitEthernet0/2 physical interface.
GigabitEthernet0/3     GigabitEthernet0/3 physical interface.
Management0/0          Management0/0 physical interface.
sensor(config-int)# physical-interfaces
```

Step 6 Specify an interface:

```
sensor(config-int)# physical-interfaces GigabitEthernet0/2
```

Step 7 Enable the interface:

```
sensor(config-int-phy)# admin-state enabled
```

The interface must be assigned to the virtual sensor (see Assigning Interfaces to the Virtual Sensor, page 5-24) and enabled to monitor traffic.

Step 8 Add a description of this interface:

```
sensor(config-int-phy)# description INT1
```

Step 9 Configure the duplex settings:

```
sensor(config-int-phy)# duplex full
```

This option is not available on modules.

Step 10 Configure the speed:

```
sensor(config-int-phy)# speed 1000
```

This option is not available on modules.

Step 11 Set up the inline VLAN pair:

```
sensor(config-int-phy)# subinterface-type inline-vlan-pair
sensor(config-int-phy-inl)# subinterface 1
sensor(config-int-phy-inl-sub)# vlan1 52
sensor(config-int-phy-inl-sub)# vlan2 53
```

Step 12 Add a description for the inline VLAN pair:

```
sensor(config-int-phy-inl-sub)# description pairs vlans 52 and 53
```

Step 13 Verify the inline VLAN pair settings:

```
sensor(config-int-phy-inl-sub)# show settings
   subinterface-number: 1
   ----------------------------------------------
      description: VLANpair1 default:
      vlan1: 52
      vlan2: 53
   ----------------------------------------------
sensor(config-int-phy-inl-sub)#
```

Step 14 Exit interface submode:

```
sensor(config-int-phy-inl-sub)# exit
sensor(config-int-phy-inl)# exit
sensor(config-int-phy)# exit
sensor(config-int)# exit
Apply Changes:?[yes]:
```

Step 15 Press **Enter** to apply the changes or enter **no** to discard them.

Inline Bypass Mode

This section describes inline bypass for sensors configured as inline interface and inline VLAN pairs, and contains the following topics:

- Understanding Inline Bypass Mode, page 5-22
- Configuring Bypass Mode, page 5-23

Understanding Inline Bypass Mode

Note For more information on using hardware bypass mode with software bypass mode, see Hardware Bypass Mode, page 5-7.

You can use inline bypass as a diagnostic tool and a failover protection mechanism. Normally, the sensor Analysis Engine performs packet analysis. When inline bypass is activated, Analysis Engine is bypassed, allowing traffic to flow through the inline interfaces and inline VLAN pairs without inspection. Inline bypass ensures that packets continue to flow through the sensor when the sensor processes are temporarily stopped for upgrades or when the sensor monitoring processes fail. There are three modes: on, off, and automatic. By default, bypass mode is set to automatic.

Caution There are security consequences when you put the sensor in bypass mode. When bypass mode is on, the traffic bypasses the sensor and is not inspected, therefore, the sensor cannot prevent malicious attacks.

Note The inline bypass functionality is implemented in software, so it only functions when the operating system is running. If the sensor is powered off or shut down, inline bypass does not work—traffic does not flow through the sensor.

Configuring Bypass Mode

Use the **bypass-mode** command in the service interface submode to configure bypass mode.

The following options apply:

- **off**—Turns off inline bypassing. Packet inspection is performed on inline data traffic. However, inline traffic is interrupted if Analysis Engine is stopped.
- **on**—Turns on inline bypassing. No packet inspection is performed on the traffic. Inline traffic continues to flow even if Analysis Engine is stopped.
- **auto**—Automatically begins bypassing inline packet inspection if Analysis Engine stops processing packets. This prevents data interruption on inline interfaces. This is the default.

Step 1 Log in to the CLI using an account with administrator privileges.

Step 2 Enter interface submode:

```
sensor# configure terminal
sensor(config)# service interface
```

Step 3 Configure bypass mode:

```
sensor(config-int)# bypass-mode off
```

Step 4 Verify the settings:

```
sensor(config-int)# show settings
-----------------------------------------------
 bypass-mode: off default: auto
 interface-notifications
 -----------------------------------------------
    missed-percentage-threshold: 0 percent <defaulted>
    notification-interval: 30 seconds <defaulted>
    idle-interface-delay: 30 seconds <defaulted>
 -----------------------------------------------
sensor(config-int)#
```

Step 5 Exit interface submode:

```
sensor(config-int)# exit
Apply Changes:?[yes]:
```

Step 6 Press **Enter** to apply the changes or enter **no** to discard them.

Assigning Interfaces to the Virtual Sensor

This section describes the virtual sensor and how to add interfaces to it. It contains the following topics:

Overview

The sensor can receive data inputs from one or many monitored data streams. These monitored data streams can either be physical interface ports or virtual interface ports. For example, a single sensor can monitor traffic from in front of the firewall and from behind the firewall. IPS 5.1 only supports one virtual sensor, so a single sensor policy and configuration are applied to all monitored data streams.

Be aware of the following limitation when adding interfaces to the sensor—the same traffic flow cannot traverse the sensor twice either through the same interface in inline mode or through separate monitored interfaces. If packets from the same traffic flow traverse the sensor twice, the virtual sensor interprets the packets as duplicates, which results in false positive alerts.

You can configure NAT to change the IP address to handle this limitation. NAT causes the sensor to treat the before and after translation packets as separate flows. For example, if a firewall is using NAT from its internal to external networks, the sensor can monitor both of these networks without problem.

You can assign interfaces, interface pairs, and VLAN pairs to the virtual sensor and you can change the description of the virtual sensor, but you cannot add a virtual sensor or change the virtual sensor name.

Configuring Interfaces for the Virtual Sensor

Use the **physical-interface** *interface_name* command in the virtual sensor submode to assign promiscuous interfaces to the virtual sensor. Use the **physical-interface** *interface_name* **subinterface-number** *subinterface_number* command in the virtual sensor submode to assign subinterfaces with inline VLAN pairs to the virtual sensor. Use the **logical-interface** *inline_interface_pair_name* command in the virtual sensor submode to assign inline interface pairs to the virtual sensor.

Make sure that you have created any inline interface pairs or inline VLAN pairs before assigning them to the virtual sensor.

To assign the interface to the virtual sensor, follow these steps:

Step 1 Log in to the CLI using an account with administrator privileges.

Step 2 Enter virtual sensor submode:

```
sensor# configure terminal
```

```
sensor(config)# service analysis-engine
sensor(config-ana)# virtual-sensor vs0
```

Step 3 Display the list of available interfaces:

```
sensor(config-ana-vir)# physical-interfaces ?
GigabitEthernet0/0     GigabitEthernet0/0 physical interface.
GigabitEthernet0/1     GigabitEthernet0/1 physical interface.
GigabitEthernet2/0     GigabitEthernet2/0 physical interface.
GigabitEthernet2/1     GigabitEthernet2/1 physical interface.
sensor(config-ana-vir)# physical-interfaces
```

Step 4 Assign the promiscuous mode interfaces to the virtual sensor:

```
sensor(config-ana-vir)# physical-interface GigabitEthernet2/0
```

Repeat Step 4 for all promiscuous mode interfaces.

Step 5 Assign the inline interface pairs to the sensor:

```
sensor(config-ana-vir)# logical-interface inline_interface_pair_name
```

Repeat Step 5 for all inline interface pair interfaces.

Step 6 Assign the subinterface with the inline VLAN pairs to the virtual sensor:

```
sensor(config-ana-vir)# physical-interface GigabitEthernet2/0 subinterface-number
subinterface_number
```

Repeat Step 6 for all subinterfaces with inline VLAN pairs.

Step 7 Exit analysis engine mode:

```
sensor(config-ana-vir)# exit
sensor(config-ana)# exit
sensor(config)#
Apply Changes:?[yes]:
```

Step 8 Press **Enter** to apply the changes or enter **no** to discard them.

Configuring Interface Notifications

You can configure the sensor to monitor the flow of packets across an interface and send notification if that flow changes (starts/stops) during a specified interval. You can configure the missed packet threshold within a specific notification interval and also configure the interface idle delay before a status event is reported.

Use the **interface-notifications** command in the service interface submode to configure traffic notifications.

The following options apply:

- **default**—Sets the value back to the system default setting.
- **idle-interface-delay**—The number of seconds an interface must be idle before sending a notification. The valid range is 5 to 3600. The default is 30 seconds.

- **missed-percentage-threshold**—The percentage of packets that must be missed during a specified interval before notification will be sent. The valid range is 0 to 100. The default is 0.
- **notification-interval**—Interval to check for missed packet percentage. The valid range is 5 to 3600. The default is 30 seconds

To configure the interface notification settings, follow these steps:

Step 1 Log in to the CLI using an account with administrator privileges.

Step 2 Enter global configuration mode:

```
sensor# configure terminal
```

Step 3 Enter interface submode:

```
sensor(config)# service interface
```

Step 4 Enter interface notifications submode:

```
sensor(config-int)# interface-notifications
```

Step 5 Configure the idle interface delay:

```
sensor(config-int-int)# idle-interface-delay 60
```

Step 6 Configure the missed percentage threshold:

```
sensor(config-int-int)# missed-percentage-threshold 1
```

Step 7 Configure the notification interval:

```
sensor(config-int-int)# notification-interval 60
```

Step 8 Verify the settings:

```
sensor(config-int-int)# show settings
   interface-notifications
   ----------------------------------------------
      missed-percentage-threshold: 1 percent default: 0
      notification-interval: 60 seconds default: 30
      idle-interface-delay: 60 seconds default: 30
   ----------------------------------------------
sensor(config-int-int)#
```

Step 9 Exit interface notifications submode:

```
sensor(config-int-int)# exit
sensor(config-int)# exit
Apply Changes:?[yes]:
```

Step 10 Press **Enter** to apply the changes or enter **no** to discard them.

CHAPTER 6

Configuring Event Action Rules

This chapter explains how to configure event action rules. It contains the following sections:

Understanding Event Action Rules

Event action rules are a group of settings you configure for the event action processing component of the sensor. These rules dictate the actions the sensor performs when an event occurs.

The event action processing component is responsible for the following functions:

- Calculating the risk rating
- Adding event action overrides
- Filtering event action
- Executing the resulting event action
- Summarizing and aggregating events
- Maintaining a list of denied attackers

Signature Event Action Processor

SEAP coordinates the data flow from the signature event in the alarm channel to processing through the SEAO, the SEAF, and the SEAH. It consists of the following components:

- Alarm channel

  The unit that represents the area to communicate signature events from the Sensor App inspection path to signature event handling.

- Signature event action override (SEAO)

  Adds actions based on the RR value. SEAO applies to all signatures that fall into the range of the configured RR threshold. Each SEAO is independent and has a separate configuration value for each action type. For more information, see Calculating the Risk Rating, page 6-8.

- Signature event action filter (SEAF)

  Subtracts actions based on the signature event's signature ID, addresses, and RR. The input to the SEAF is the signature event with actions possibly added by the SEAO.

  Note The SEAF can only subtract actions, it cannot add new actions.

  The following parameters apply to the SEAF:

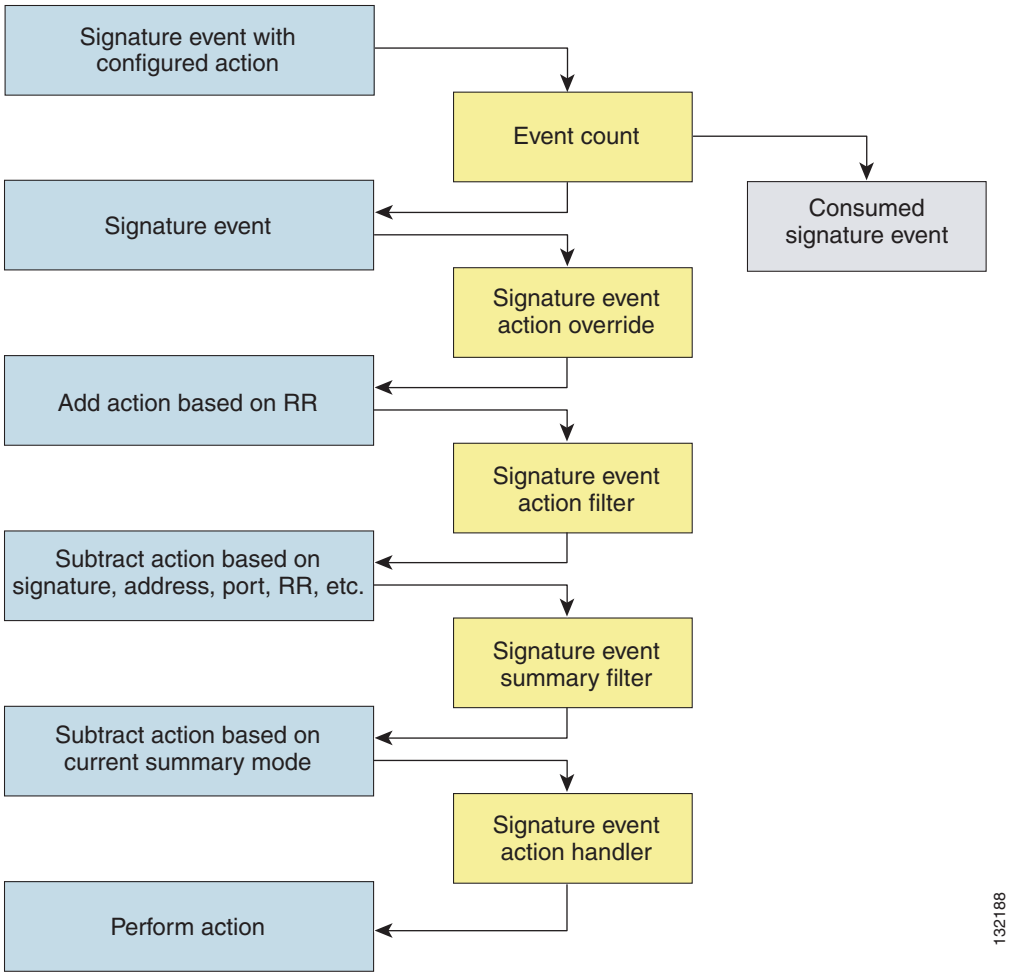
  - Signature ID
  - Subsignature ID
  - Attacker address
  - Attacker port
  - Victim address
  - Victim port
  - RR threshold range
  - Actions to subtract
  - Sequence identifier (optional)
  - Stop-or-continue bit
  - Enable action filter line bit

- Signature event action handler (SEAH)

  Performs the requested actions. The output from the SEAH is the actions being performed and possibly an evIdsAlert written to the Event Store.

Figure 6-1 on page 6-3 illustrates the logical flow of the signature event through the SEAP and the operations performed on the action for this event. It starts with the signature event with configured action received in the alarm channel and flows top to bottom as the signature event passes through the functional components of the SEAP.

Figure 6-1 Signature Event Through SEAP



Event Actions

Table 6-1 describes the event actions.

Table 6-1 Event Actions

Event Action Name	Description
Deny Attacker Inline	(Inline mode only) Does not transmit this packet and future packets originating from the attacker address for a specified period of time.[1] **Note** This is the most severe of the deny actions. It denies current and future packets from a single attacker address. You can clear all denied attacker entries, which permits the addresses back on the network. For the procedure, see Monitoring and Clearing the Denied Attackers List, page 6-21.
Deny Attacker Service Pair Inline	(Inline mode only) Does not transmit this packet and future packets on the attacker address victim port pair for a specified period of time.
Deny Attacker Victim Pair Inline	(Inline mode only) Does not transmit this packet and future packets on the attacker/victim address pair for a specified period of time. **Note** For deny actions, you can set the specified period of time and maximum number of denied attackers. For the procedure, see Configuring the General Settings, page 6-20.
Deny Connection Inline	(Inline mode only) Does not transmit this packet and future packets on the TCP flow.
Deny Packet Inline	(Inline mode only) Does not transmit this packet. **Note** You cannot delete the event action override for Deny Packet Inline because it is protected. If you do not want to use that override, disable it.
Log Attacker Packets	Starts IP logging packets containing the attacker address. **Note** This action causes an alert to be written to the Event Store, even if Produce Alert is not selected.
Log Pair Packets	Starts IP logging packets containing the attacker-victim address pair. **Note** This action causes an alert to be written to the Event Store, even if Produce Alert is not selected.
Log Victim Packets	Starts IP logging packets containing the victim address.
Modify Packet Inline	Modifies packet data to remove ambiguity about what the end point might do with the packet. **Note** Modify Packet Inline is not an option for Add Event Action Filter or Add Event Action Override.

Table 6-1 Event Actions (continued)

Event Action Name	Description
Produce Alert	Writes the event to the Event Store as an alert. **Note** The Produce Alert action is not automatic when you enable alerts for a signature. To have an alert created in the Event Store, you must select Produce Alert. If you add a second action, you must include Produce Alert if you want an alert sent to the Event Store. Also, every time you configure the event actions, a new list is created and it replaces the old list. Make sure you include all the event actions you need for each signature.
Produce Verbose Alert	Includes an encoded dump of the offending packet in the alert. **Note** This action causes an alert to be written to the Event Store, even if Produce Alert is not selected.
Request Block Connection	Sends a request to ARC to block this connection. **Note** You must have blocking devices configured to implement this action. For more information, see Chapter 10, "Configuring Attack Response Controller for Blocking and Rate Limiting."
Request Block Host	Sends a request to ARC to block this attacker host. **Note** You must have blocking devices configured to implement this action. For more information, see Chapter 10, "Configuring Attack Response Controller for Blocking and Rate Limiting." **Note** For block actions, you can set the duration of the block. For the procedure, see Configuring the General Settings, page 6-20.
Request Rate Limit	Sends a rate limit request to ARC to perform rate limiting. You must have rate limiting devices configured to implement this action. For more information, see Chapter 10, "Configuring Attack Response Controller for Blocking and Rate Limiting."
Request SNMP Trap	Sends a request to NotificationApp to perform SNMP notification. **Note** This action causes an alert to be written to the Event Store, even if Produce Alert is not selected. You must have SNMP configured on the sensor to implement this action. For more information, see Chapter 11, "Configuring SNMP."
Reset TCP Connection	Sends TCP resets to hijack and terminate the TCP flow. **Note** Reset TCP Connection only works on TCP signatures that analyze a single connection. It does not work for sweeps or floods.

1. The sensor maintains a list of attackers being denied by the system. To remove an entry from the denied attacker list, you can view the list of attackers and clear the entire list, or you can wait for the timer to expire. The timer is a sliding timer for each entry. Therefore, if attacker A is being denied, but issues another attack, the timer for attacker A is reset and attacker A remains in the denied attacker list until the timer expires. If the denied attacker list is at capacity and cannot add a new entry, the packet will still be denied.

Understanding Deny Packet Inlin

For signatures that have deny-packet-inline configured as an action or for an event action override that adds deny-packet-inline as an action, the following actions may be taken:

- droppedPacket
- deniedFlow
- tcpOneWayResetSent

The deny packet inline action is represented as a dropped packet action in the alert. When a deny packet inline occurs for a TCP connection, it is automatically upgraded to a deny connection inline action and seen as a denied flow in the alert. If the IPS denies just one packet, the TCP continues to try to send that same packet again and again, so the IPS denies the entire connection to ensure it never succeeds with the resends.

When a deny connection inline occurs, the IPS also automatically sends a TCP one-way reset, which shows up as a TCP one-way reset sent in the alert. When the IPS denies the connection, it leaves an open connection on both the client (generally the attacker) and the server (generally the victim). Too many open connections can result in resource problems on the victim. So the IPS sends a TCP reset to the victim to close the connection on the victim side (usually the server), which conserves the resources of the victim. It also prevents a failover that would otherwise allow the connection to fail over to a different network path and reach the victim. The IPS leaves the attacker side open and denies all traffic from it.

Task List for Configuring Event Action Rules

Follow these steps when configuring the event action rules component of the IPS:

1. Create any variables that you want to use in event action filters.
2. Create TVRs.

   Assign TVRs to your network assets so that you can calculate the RR.
3. Create overrides to add actions based on the RR value.

   Assign an RR to each event action type.
4. Create filters.

   Assign filters to subtract actions based on the signature's ID, IP addresses, and RR.
5. Configure the general settings.

   Specify whether you want to use the summarizer, the meta event generator, or configure denied attacker parameters.

Event Action Variables

This section describes event action variables, and contains the following topics:

Understanding Event Action Variables

You can create event action variables and then use those variables in event action filters. When you want to use the same value within multiple filters, use a variable. When you change the value of the variable, any filter that uses that variable is updated with the new value.

Note You must preface the variable with a dollar ($) sign to indicate that you are using a variable rather than a string.

Some variables cannot be deleted because they are necessary to the signature system. If a variable is protected, you cannot edit it. You receive an error message if you try to delete protected variables. You can edit only one variable at a time.

When configuring IP addresses, specify the full IP address or ranges or set of ranges. For example:

- 10.90.1.1
- 10.89.10.10-10.89.10.23
- 10.1.1.1-10.2.255.255, 10.89.10.10-10.89.10.23

Timesaver For example, if you have an IP address space that applies to your engineering group and there are no Windows systems in that group, and you are not worried about any Windows-based attacks to that group, you could set up a variable to be the engineering group's IP address space. You could then use this variable to configure a filter that would ignore all Windows-based attacks for this group.

Configuring Event Action Variables

Use the **variables** *variable_name* **address** *ip_address* command in service event action rules submode to set up event action variables. The IP address can be one address, a range, or ranges separated by a comma.

To configure event action variables, follow these steps:

Step 1 Log in to the CLI using an account with administrator privileges.

Step 2 Enter event action rules submode:

```
sensor# configure terminal
sensor(config)# service event-action-rules rules0
```

Step 3 Create a variable:

```
sensor(config-rul)# variables variable1 address 10.89.130.108
```

The valid values for **address** are A.B.C.D-A.B.C.D [,A.B.C.D-A.B.C.D].

Step 4 Check the variable you just made:

```
sensor(config-rul)# show settings
   variables (min: 0, max: 256, current: 2)
   ----------------------------------------------
      variableName: variable1
      ----------------------------------------------
         address: 10.89.130.108 default: 0.0.0.0-255.255.255.255
      ----------------------------------------------
```

```
-----------------------------------------------
```

Step 5 Exit event action rules submode:

```
sensor(config-rul)# exit
Apply Changes:?[yes]:
```

Step 6 Press **Enter** to apply your changes or enter **no** to discard them.

Target Value Ratings

This section describes what RR is and how to use it to configure the TVR. This section contains the following topics:

- Calculating the Risk Rating, page 6-8
- Configuring the Target Value Rating, page 6-9

Calculating the Risk Rating

An RR is a value between 0 and 100 that represents a numerical quantification of the risk associated with a particular event on the network. The calculation takes into account the value of the network asset being attacked (for example, a particular server), so it is configured on a per-signature basis (ASR and SFR) and on a per-server basis (TVR).

RRs let you prioritize alerts that need your attention. These RR factors take into consideration the severity of the attack if it succeeds, the fidelity of the signature, and the overall value of the target host to you. The RR is reported in the evIdsAlert.

The following values are used to calculate the RR for a particular event:

- **Attack Severity Rating (ASR)**—A weight associated with the severity of a successful exploit of the vulnerability.

  The ASR is derived from the alert severity parameter of the signature.

- **Signature Fidelity Rating (SFR)**—A weight associated with how well this signature might perform in the absence of specific knowledge of the target.

  SFR is calculated by the signature author on a per-signature basis. The signature author defines a baseline confidence ranking for the accuracy of the signature in the absence of qualifying intelligence on the target. It represents the confidence that the detected behavior would produce the intended effect on the target platform if the packet under analysis were allowed to be delivered. For example, a signature that is written with very specific rules (specific regular expression) has a higher SFR than a signature that is written with generic rules.

- **Target Value Rating (TVR)**—A weight associated with the perceived value of the target.

  TVR is a user-configurable value that identifies the importance of a network asset (through its IP address). You can develop a security policy that is more stringent for valuable corporate resources and looser for less important resources. For example, you could assign a TVR to the company web server that is higher than the TVR you assign to a desktop node. In this example, attacks against the company web server have a higher RR than attacks against the desktop node.

Note RR is a product of ASR, SFR, and TVR with an optional PD (promiscuous delta) subtracted in promiscuous mode only.

Configuring the Target Value Rating

You can assign a TVR to your network assets. T he TVR is one of the factors used to calculate the RR value for each alert. You can assign different TVRs to different targets. Events with a higher RR trigger more severe signature event actions.

Use the **target-value target-value-setting** {**zerovalue** | **low** | **medium** | **high** | **mission-critical**} **target-address** *ip_address* command in service event action rules submode to set TVRs for your network assets. The default is medium.

The following options apply:

- **target-address** *ip_address*—Range set of IP address(es).
- **target-value-setting**—Choose one of the following:
  - **zerovalue**—No value of this target.
  - **low**—Lower value of this target.
  - **medium**—Normal value of this target.
  - **high**—Elevated value of this target.
  - **mission-critical**—Extreme value of this target.

To configure TVRs for your network assets, follow these steps:

Step 1 Log in to the CLI using an account with administrator privileges.

Step 2 Enter event action rules submode:

```
sensor# configure terminal
sensor(config)# service event-action-rules rules0
```

Step 3 Assign the TVR to the network asset:

```
sensor(config-rul)# target-value target-value-setting mission-critical target-address
10.89.130.108
```

Step 4 Check the TVR setting you just configured:

```
sensor(config-rul)# show settings
----------------------------------------------
   target-value (min: 0, max: 5, current: 1)
   ----------------------------------------------
      target-value-setting: mission-critical
      target-address: 10.89.130.108 default: 0.0.0.0-255.255.255.255
   ----------------------------------------------
sensor(config-rul)#
```

Step 5 Exit event action rules submode:

```
sensor(config-rul)# exit
Apply Changes:?[yes]:
```

Step 6 Press **Enter** to apply your changes or enter `no` to discard them.

Event Action Overrides

This section describes event action overrides, and contains the following topics:

Understanding Event Action Overrides

You can add an event action override to change the actions associated with an event based on the RR of that event. Event action overrides are a way to add event actions globally without having to configure each signature individually. Each event action has an associated RR range. If a signature event occurs and the RR for that event falls within the range for an event action, that action is added to the event. For example, if you want any event with an RR of 85 or more to generate an SNMP trap, you can set the RR range for Request SNMP Trap to 85-100. If you do not want to use action overrides, you can disable the entire event action override component.

Understanding Deny Packet Inline

For signatures that have deny-packet-inline configured as an action or for an event action override that adds deny-packet-inline as an action, the following actions may be taken:

- droppedPacket
- deniedFlow
- tcpOneWayResetSent

The deny packet inline action is represented as a dropped packet action in the alert. When a deny packet inline occurs for a TCP connection, it is automatically upgraded to a deny connection inline action and seen as a denied flow in the alert. If the IPS denies just one packet, the TCP continues to try to send that same packet again and again, so the IPS denies the entire connection to ensure it never succeeds with the resends.

When a deny connection inline occurs, the IPS also automatically sends a TCP one-way reset, which shows up as a TCP one-way reset sent in the alert. When the IPS denies the connection, it leaves an open connection on both the client (generally the attacker) and the server (generally the victim). Too many open connections can result in resource problems on the victim. So the IPS sends a TCP reset to the victim to close the connection on the victim side (usually the server), which conserves the resources of the victim. It also prevents a failover that would otherwise allow the connection to fail over to a different network path and reach the victim. The IPS leaves the attacker side open and denies all traffic from it.

Configuring Event Action Overrides

Use the **overrides** {**request-block-connection** | **request-block-host** | **deny-attacker-inline** | **deny-packet-inline** | **deny-attacker-service-pair-inline** | **deny-attacker-victim-pair-inline** | **deny-connection-inline** | **log-attacker-packets** | **log-victim-packets** | **log-pair-packets** | **reset-tcp-connection** | **produce-alert** | **produce-verbose-alert** | **request-rate-limit** | **request-snmp-trap**} command in service event action rules submode to configure the parameters of event action overrides.

For a description of all the event actions, see Event Actions, page 6-4.

Note You cannot delete the event action override for **deny-packet-inline** because it is protected. If you do not want to use that override, disable it.

The following options apply:

- **no**—Removes an entry or selection setting.
- **override-item-status** {**enabled** | **disabled**}—Enables or disables the use of this override item. The default is enabled.
- **risk-rating-range**—Range of RR values for this override item. The default is 1 to 100.
- **show**—Displays system settings and/or history information.

To add event action overrides, follow these steps:

Step 1 Log in to the CLI using an account with administrator privileges.

Step 2 Enter event action rules submode:

```
sensor# configure terminal
sensor(config)# service event-action-rules rules0
sensor(config-rul)#
```

Step 3 To configure how packets are treated for overrides:

a. To deny packets from the source IP address of the attacker:

```
sensor(config-rul)# overrides deny-attacker-inline
sensor(config-rul-ove)#
```

b. To not transmit the single packet causing the alert:

```
sensor(config-rul-ove)# exit
sensor(config-rul)# overrides deny-packet-inline
sensor(config-rul-ove)#
```

c. To not transmit packets on the specified TCP connection:

```
sensor(config-rul-ove)# exit
sensor(config-rul)# overrides deny-connection-inline
sensor(config-rul-ove)#
```

d. To send TCP RST packets to terminate the connection:

```
sensor(config-rul-ove)# exit
sensor(config-rul)# overrides reset-tcp-connection
sensor(config-rul-ove)#
```

Step 4 To configure overrides to request blocks:

a. To request a block of the connection:

```
sensor(config-rul-ove)# exit
sensor(config-rul)# overrides request-block-connection
sensor(config-rul-ove)#
```

b. To request a block of the attacker host:

```
sensor(config-rul-ove)# exit
sensor(config-rul)# overrides request-block-host
sensor(config-rul-ove)#
```

Step 5 To log packets for overrides:

a. To log the packets from the attacker IP address:

```
sensor(config-rul-ove)# exit
sensor(config-rul)# overrides log-attacker-packets
sensor(config-rul-ove)#
```

b. To log the packets from the victim IP address:

```
sensor(config-rul-ove)# exit
sensor(config-rul)# overrides log-victim-packets
sensor(config-rul-ove)#
```

c. To log packets from both the attacker and victim IP addresses:

```
sensor(config-rul-ove)# exit
sensor(config-rul)# overrides log-pair-packets
sensor(config-rul-ove)#
```

Step 6 To write alerts to the Event Store:

a. To write an alert to the Event Store:

```
sensor(config-rul-ove)# exit
sensor(config-rul)# overrides produce-alert
sensor(config-rul-ove)#
```

b. To write verbose alerts to the Event Store:

```
sensor(config-rul-ove)# exit
sensor(config-rul)# overrides produce-verbose-alert
sensor(config-rul-ove)#
```

c. To write events that request an SNMP trap to the Event Store:

```
sensor(config-rul-ove)# exit
sensor(config-rul)# overrides request-snmp-trap
sensor(config-rul-ove)#
```

Step 7 To configure the RR for this override item:

```
sensor(config-rul-ove)# risk-rating-range 85-100
```

Note The default RR range is 0 to 100. Set it to a different value, such as 85 to 100.

Step 8 To enable or disable the use of this override item:

```
sensor(config-rul-ove)# override-item-status {enabled | disabled}
```

The default is enabled.

Step 9 Verify the settings:

```
sensor(config-rul-ove)# show settings
   action-to-add: deny-attacker-inline default: produce-alert
   ----------------------------------------------
      override-item-status: Enabled default: Enabled
      risk-rating-range: 85-100 default: 0-100
   ----------------------------------------------
```

Step 10 Exit event action rules submode:

```
sensor(config-rul-ove)# exit
sensor(config-rul)#
Apply Changes:?[yes]:
```

Step 11 Press **Enter** to apply your changes or enter **no** to discard them.

Event Action Filters

This section describes event action filters. and contains the following topics:

- Understanding Event Action Filters, page 6-13
- Configuring Event Action Filters, page 6-13

Understanding Event Action Filters

Event action filters are processed as an ordered list and you can move filters up or down in the list. Filters let the sensor perform certain actions in response to the event without requiring the sensor to perform all actions or remove the entire event. Filters work by removing actions from an event. A filter that removes all actions from an event effectively consumes the event.

Note When filtering sweep signatures, we recommend that you do not filter the destination addresses. If there are multiple destination addresses, only the last address is used for matching the filter.

Caution Event action filters based on source and destination IP addresses do not function for the Sweep engine, because they do not filter as regular signatures. To filter source and destination IP addresses in sweep alerts, use the source and destination IP address filter parameters in the Sweep engine signatures.

Configuring Event Action Filters

You can configure event action filters to remove specific actions from an event or to discard an entire event and prevent further processing by the sensor. You can use event action variables that you defined to group addresses for your filters. For the procedure for configuring event action variables, see Configuring Event Action Variables, page 6-7.

Note You must preface the variable with a dollar sign ($) to indicate that you are using a variable rather than a string. Otherwise, you receive the `Bad source and destination` error.

Use the **filters {dit | insert | move] name1 [begin | end | inactive | before | after}** command in service event action rules submode to set up event action filters.

The following options apply:

- **actions-to-remove**—Event actions to remove for this filter item.
  - **deny-attacker-inline** —(Inline mode only) does not transmit this packet and future packets from the attacker address for a specified period of time.
  - **deny-attacker-service-pair-inline**—(Inline only) Does not transmit this packet and future packets on the attacker address victim port pair for a specified period of time.
  - **deny-attacker-victim-pair-inline**—(Inline only) Does not transmit this packet and future packets on the attacker/victim address pair for a specified period of time.
  - **deny-connection-inline**—(Inline mode only) Does not transmit this packet and future packets on the TCP Flow.
  - **deny-packet-inline**—(Inline mode only) Does not transmit this packet.
  - **log-attacker-packets**—Starts IP logging of packets containing the attacker address. This action causes an alert to be written to the Event Store, even if **produce-alert** is not selected.
  - **log-pair-packets**—Starts IP logging of packets containing the attacker-victim address pair. This action causes an alert to be written to the Event Store, even if **produce-alert** is not selected.
  - **log-victim-packets**—Starts IP logging of packets containing the victim address. This action causes an alert to be written to the Event Store, even if **produce-alert** is not selected.
  - **produce-alert** —Writes the event to the Event Store as an alert.
  - **produce-verbose-alert**—Includes an encoded dump (possibly truncated) of the offending packet in the alert. This action causes an alert to be written to the Event Store, even if **produce-alert** is not selected.
  - **request-block-connection**—Sends a request to ARC to block this connection. You must have blocking devices configured to implement this action.
  - **request-block-host**—Sends a request to ARC to block this attacker host. You must have blocking devices configured to implement this action.
  - **request-rate-limit**—Sends a rate limit request to ARC to perform rate limiting. You must have rate limiting devices configured to implement this action.
  - **request-snmp-trap**—Sends a request to the Notification Application component of the sensor to perform SNMP notification. This action causes an alert to be written to the Event Store, even if **produce-alert** is not selected. You must have SNMP configured on the sensor to implement this action.
  - **reset-tcp-connection**—Sends TCP resets to hijack and terminate the TCP flow. **Reset TCP Connection** only works on TCP signatures that analyze a single connection. It does not work for sweeps or floods.
  - **modify-packet-inline**— Modifies packet data to remove ambiguity about what the end point might do with the packet.
- **attacker-address-range**—Range set of attacker address(es) for this item (for example, 10.20.1.0-10.20.1.255,10.20.5.0-10.20.5.255).
- **attacker-port-range**—Range set of attacker port(s) for this item (for example, 147-147,8000-10000).
- **default**—Sets the value back to the system default setting.

- **deny-attacker-percentage**—Percentage of packets to deny for deny attacker features. The valid range is 0 to 100. The default is 100.
- **filter-item-status** {**enabled** | **disabled**}—Enables or disables the use of this filter item.
- **no**—Removes an entry or selection setting.
- **risk-rating-range**—Range of RR values for this filter item.
- **signature-id-range**—Range set of signature ID(s) for this item (for example, 1000-2000,3000-3000).
- **stop-on-match**—Continues evaluating filters or stops when this filter item is matched.
- **subsignature-id-range**—Range set of subsignature ID(s) for this item (for example, 0-2,5-5).
- **user-comment** —Lets you add your comments about this filter item.
- **victim-address-range**—Range set of victim address(es) for this item (for example, 10.20.1.0-10.20.1.255,10.20.5.0-10.20.5.255).
- **victim-port-range**—Range set of victim port(s) for this item (for example, 147-147,8000-10000).

To configure event action filters, follow these steps:

Step 1 Log in to the CLI using an account with administrator privileges.

Step 2 Enter event action rules submode:

```
sensor# configure terminal
sensor(config)# service event-action-rules rules0
sensor(config-rul)#
```

Step 3 Create the filter name:

```
sensor(config-rul)# filters insert name1 begin
```

Use **name1**, **name2**, and so forth to name your event action filters. Use the **begin | end | inactive | before | after** keywords to specify where you want to insert the filter.

Step 4 Configure the values for this filter:

a. Set the signature ID range:

```
sensor(config-rul-fil)# signature-id-range 1000-1005
```

The default is 900 to 65535.

b. Set the subsignature ID range:

```
sensor(config-rul-fil)# subsignature-id-range 1-5
```

The default is 0 to 255.

c. Set the attacker address range:

```
sensor(config-rul-fil)# attacker-address-range 10.89.10.10-10.89.10.23
```

The default is 0.0.0.0 to 255.255.255.255.

d. Set the victim address range:

```
sensor(config-rul-fil)# victim-address-range 192.56.10.1-192.56.10.255
```

The default is 0.0.0.0 to 255.255.255.255.

e. Set the victim port range:

```
sensor(config-rul-fil)# victim-port-range 0-434
```

The default is 0 to 65535.

f. Set the risk rating range:

```
sensor(config-rul-fil)# risk-rating-range 85-100
```

The default is 0 to 100.

g. Set the actions to remove:

```
sensor(config-rul-fil)# actions-to-remove reset-tcp-connection
```

h. If you are filtering a deny action, set the percentage of deny actions you want:

```
sensor(config-rul-fil)# deny-attacker-percentage 90
```

The default is 100.

i. Set the status of the filter to either disabled or enabled.

```
sensor(config-rul-fil)# filter-item-status {enabled | disabled}
```

The default is enabled.

j. Set the stop on match parameter.

```
sensor(config-rul-fil)# stop-on-match {true | false}
```

True tells the sensor to stop processing filters if this item matches. **False** tells the sensor to continue processing filters even if this item matches.

k. Add any comments you want to explain this filter:

```
sensor(config-rul-fil)# user-comments
```

Step 5 Verify the settings for the filter:

```
sensor(config-rul-fil)# show settings
   NAME: name1
   ----------------------------------------------
      signature-id-range: 1000-10005 default: 900-65535
      subsignature-id-range: 1-5 default: 0-255
      attacker-address-range: 10.89.10.10-10.89.10.23 default: 0.0.0.0-255.255.255.255
      victim-address-range: 192.56.10.1-192.56.10.255 default: 0.0.0.0-255.255.255.255
      attacker-port-range: 0-65535 <defaulted>
      victim-port-range: 1-343 default: 0-65535
      risk-rating-range: 85-100 default: 0-100
      actions-to-remove: reset-tcp-connection default:
      deny-attacker-percentage: 90 default: 100
      filter-item-status: Enabled default: Enabled
      stop-on-match: True default: False
      user-comment: This is a new filter. default:
   ----------------------------------------------
ssenor(config-rul-fil)#
```

Step 6 To edit an existing filter:

```
sensor(config-rul)# filters edit name1
```

Step 7 Edit the parameters (see Steps 4a through 4k).

Step 8 To move a filter up or down in the filter list:

```
sensor(config-rul-fil)# exit
sensor(config-rul)# filters move name5 before name1
```

Step 9 Verify that you have moved the filters:

```
sensor(config-rul-fil)# exit
sensor(config-rul)# show settings
   ----------------------------------------------
   filters (min: 0, max: 4096, current: 5 - 4 active, 1 inactive)
   ----------------------------------------------
   ACTIVE list-contents
   ----------------------------------------------
      NAME: name5
      ----------------------------------------------
         signature-id-range: 900-65535 <defaulted>
         subsignature-id-range: 0-255 <defaulted>
         attacker-address-range: 0.0.0.0-255.255.255.255 <defaulted>
         victim-address-range: 0.0.0.0-255.255.255.255 <defaulted>
         attacker-port-range: 0-65535 <defaulted>
         victim-port-range: 0-65535 <defaulted>
         risk-rating-range: 0-100 <defaulted>
         actions-to-remove: <defaulted>
         filter-item-status: Enabled <defaulted>
         stop-on-match: False <defaulted>
         user-comment: <defaulted>
      ----------------------------------------------
      ----------------------------------------------
      NAME: name1
      ----------------------------------------------
         signature-id-range: 900-65535 <defaulted>
         subsignature-id-range: 0-255 <defaulted>
         attacker-address-range: 0.0.0.0-255.255.255.255 <defaulted>
         victim-address-range: 0.0.0.0-255.255.255.255 <defaulted>
         attacker-port-range: 0-65535 <defaulted>
         victim-port-range: 0-65535 <defaulted>
         risk-rating-range: 0-100 <defaulted>
         actions-to-remove: <defaulted>
         filter-item-status: Enabled <defaulted>
         stop-on-match: False <defaulted>
         user-comment: <defaulted>
      ----------------------------------------------
      ----------------------------------------------
      NAME: name2
      ----------------------------------------------
         signature-id-range: 900-65535 <defaulted>
         subsignature-id-range: 0-255 <defaulted>
         attacker-address-range: 0.0.0.0-255.255.255.255 <defaulted>
         victim-address-range: 0.0.0.0-255.255.255.255 <defaulted>
         attacker-port-range: 0-65535 <defaulted>
         victim-port-range: 0-65535 <defaulted>
         risk-rating-range: 0-100 <defaulted>
         actions-to-remove: <defaulted>
         filter-item-status: Enabled <defaulted>
         stop-on-match: False <defaulted>
         user-comment: <defaulted>
      ----------------------------------------------
      ----------------------------------------------
   ----------------------------------------------
   INACTIVE list-contents
   ----------------------------------------------
----------------------------------------------
sensor(config-rul)#
```

Step 10 To move a filter to the inactive list:

```
sensor(config-rul)# filters move name1 inactive
```

Step 11 Verify that the filter has been moved to the inactive list:

```
sensor(config-rul-fil)# exit
sensor(config-rul)# show settings
   ----------------------------------------------
   INACTIVE list-contents
   ----------------------------------------------
      ------------------------------------------------
      NAME: name1
      ------------------------------------------------
         signature-id-range: 900-65535 <defaulted>
         subsignature-id-range: 0-255 <defaulted>
         attacker-address-range: 0.0.0.0-255.255.255.255 <defaulted>
         victim-address-range: 0.0.0.0-255.255.255.255 <defaulted>
         attacker-port-range: 0-65535 <defaulted>
         victim-port-range: 0-65535 <defaulted>
         risk-rating-range: 0-100 <defaulted>
         actions-to-remove: <defaulted>
         filter-item-status: Enabled <defaulted>
         stop-on-match: False <defaulted>
         user-comment: <defaulted>
      ------------------------------------------------
      ------------------------------------------------
sensor(config-rul)#
```

Step 12 Exit event action rules submode:

```
sensor(config-rul)# exit
Apply Changes:?[yes]:
```

Step 13 Press **Enter** to apply your changes or enter **no** to discard them.

General Settings

This section describes the general settings, and contains the following topics:

Understanding the General Settings

You can configure the general settings that apply to event action rules, such as whether you want to use the summarizer and the meta event generator. The summarizer groups events into a single alert, thus decreasing the number of alerts the sensor sends out. The meta event generator processes the component events, which lets the sensor watch for suspicious activity transpiring over a series of events.

You can configure how long you want to deny attackers, the maximum number of denied attackers, and how long you want blocks to last.

Understanding Event Action Summarization

Summarization decreases the volume of alerts sent out from the sensor by providing basic aggregation of events into a single alert. Special parameters are specified for each signature and they influence the handling of the alerts. Each signature is created with defaults that reflect a preferred normal behavior. However, you can tune each signature to change this default behavior within the constraints for each engine type.

The non-alert generating actions (deny, block, TCP reset) go through the filters for each signature event unsummarized. The alert-generating actions are not performed on these summarized alerts; instead the actions are applied to the one summary alert and then put through the filters.

If you select one of the other alert-generating actions and do not have it filtered out, the alert is created even if you do not select Produce Alert. To prevent alerts from being created, you must have all alert-generating actions filtered out.

Summarization and event actions are processed after the Meta engine has processed the component events. This lets the sensor watch for suspicious activity transpiring over a series of events.

Understanding Event Action Aggregation

Basic aggregation provides two operating modes. The simple mode involves configuring a threshold number of hits for a signature that must be met before the alert is sent. A more advanced mode is timed-interval counting. In this mode, the sensor tracks the number of hits per second and only sends alerts when that threshold is met. In this example, a *hit* is a term used to describe an event, which is basically an alert, but it is not sent out of the sensor as an alert until the threshold number of hits has been exceeded.

You can select from the following summarization options:

- Fire All—Fire All mode fires an alert each time the signature is triggered. If the threshold is set for summarization, the following happens: Alerts are fired for each execution until summarization occurs. After summarization starts only one alert every summary interval fires for each address set. Alerts for other address sets are either all seen or separately summarized. The signature reverts to Fire All mode after a period of no alerts of that signature.
- Summary—Summary mode fires an alert the first time a signature is triggered, and then additional alerts for that signature are summarized for the duration of the summary interval. Only one alert every summary interval should fire for each address set. If the global summary threshold is reached, the signature goes into Global Summarization mode.
- Global Summarization—Global Summarization mode fires an alert for every summary interval. Signatures can be preconfigured for global summarization.
- Fire Once—Fire Once mode fires an alert for each address set. You can upgrade this mode to Global Summarization mode.

Understanding the Deny Attackers Inline Event Action

You can configure certain aspects of the deny attackers inline event action. You can configure the number of seconds you want to deny attackers inline and you can limit the number of attackers you want denied in the system at any one time.

Configuring the General Settings

Use the following commands in service event action rules submode to configure general event action rules settings:

- **global-block-timeout** —Number of minutes to block a host or connection.

  The valid range is 0 to 10000000. The default is 30 minutes.
- **global-deny-timeout**—Number of seconds to deny attackers inline.

  The valid range is 0 to 518400. The default is 3600.
- **global-filters-status** {**enabled | disabled**}—Enables or disables the use of the filters.

  The default is enabled.
- **global-metaevent-status {enabled | disabled}**—Enables or disables the use of the Meta Event Generator.

  The default is enabled.
- **global-overrides-status** {**enabled | disabled**}—Enables or disables the use of the overrides.

  The default is enabled.
- **global-summarization-status {enabled | disabled}**—Enables or disables the use of the summarizer.

  The default is enabled.
- **max-denied-attackers**—Limits the number of denied attackers possible in the system at any one time.

  The valid range is 0 to 100000000. The default is 10000.

To configure event action general settings, follow these steps:

Step 1 Log in to the CLI using an account with administrator privileges.

Step 2 Enter event action rules submode:

```
sensor# configure terminal
sensor(config)# service event-action-rules rules0
```

Step 3 Enter general submode:

```
sensor(config)# general
```

Step 4 To enable or disable the meta event generator:

```
sensor(config-rul-gen)# global-metaevent-status {enabled | disabled}
```

The default is enabled.

Step 5 To enable or disable the summarizer:

```
sensor(config-rul-gen)# global-summarization-status {enabled | disabled}
```

The default is enabled.

Step 6 To configure the denied attackers inline event action:

a. To limit the number of denied attackers in the system at any given time:

```
sensor(config-rul-gen)# max-denied-attackers 100
```

The default is 1000.

b. To configure the amount of seconds to deny attackers in the system:

```
sensor(config-rul-gen)# global-deny-timeout 1000
```

The default is 3600 seconds.

Step 7 To configure the number of minutes to block a host or a connection:

```
sensor(config-rul-gen)# global-block-timeout 20
```

The default is 30 minutes.

Step 8 To enable or disable any overrides that you have set up:

```
sensor(config-rul-gen)# global-overrides-status {enabled | disabled}
```

The default is enabled.

Step 9 To enable or disable any filters that you have set up:

```
sensor(config-rul-gen)# global-filters-status {enabled | disabled}
```

The default is enabled.

Step 10 Check the settings for general submode:

```
sensor(config-rul-gen)# show settings
   general
   ----------------------------------------------
      global-overrides-status: Enabled default: Enabled
      global-filters-status: Enabled default: Enabled
      global-summarization-status: Enabled default: Enabled
      global-metaevent-status: Enabled default: Enabled
      global-deny-timeout: 1000 default: 3600
      global-block-timeout: 20 default: 30
      max-denied-attackers: 100 default: 10000
   ----------------------------------------------
sensor(config-rul-gen)#
```

Step 11 Exit event action rules submode:

```
sensor(config-rul-gen)# exit
sensor(config-rul)# exit
Apply Changes:?[yes]:
```

Step 12 Press **Enter** to apply your changes or enter **no** to discard them.

Monitoring and Clearing the Denied Attackers List

Use the **show statistics denied-attackers** command to display the list of denied attackers. Use the **clear denied-attackers** command in service event action rules submode to delete the denied attackers list and clear the virtual sensor statistics.

If your sensor is configured to operate in inline mode, the traffic is passing through the sensor. You can configure signatures to deny packets, connections, and attackers while in inline mode, which means that single packets, connections, and specific attackers will be denied, that is, not transmitted, when the sensor encounters them.

When the signature fires, the attacker is denied and placed in a list. As part of sensor administration, you may want to delete the list or clear the statistics in the list.

To display the list of denied attackers and delete the list and clear the statistics, follow these steps:

Step 1 Log in to the CLI using an account with administrator privileges.

Step 2 Display the list of denied IP addresses:

```
sensor# show statistics denied-attackers
Denied Attackers and hit count for each.
   10.20.4.2 = 9
   10.20.5.2 = 5
```

The statistics show that there are two IP addresses being denied at this time.

Step 3 Delete the denied attackers list:

```
sensor# clear denied-attackers
Warning: Executing this command will delete all addresses from the list of
attackers currently being denied by the sensor.
Continue with clear? [yes]:
```

Step 4 Enter **yes** to clear the list.

Step 5 Verify that you have cleared the list:

```
sensor# show statistics virtual-sensor
Virtual Sensor Statistics
   Statistics for Virtual Sensor vs0
      Name of current Signature-Definition instance = sig0
      Name of current Event-Action-Rules instance = rules0
      List of interfaces monitored by this virtual sensor = mypair
      Denied Address Information
         Number of Active Denied Attackers = 0
         Number of Denied Attackers Inserted = 2
         Number of Denied Attackers Total Hits = 287
         Number of times max-denied-attackers limited creation of new entry = 0
         Number of exec Clear commands during uptime = 1
      Denied Attackers and hit count for each.
```

There is no longer any information under the `Denied Attackers and hit count for each category`.

Step 6 To clear only the statistics:

```
sensor# show statistics virtual-sensor clear
```

Step 7 Verify that you have cleared the statistics:

```
JWK-4255# show statistics virtual-sensor
Virtual Sensor Statistics
   Statistics for Virtual Sensor vs0
      Name of current Signature-Definition instance = sig0
      Name of current Event-Action-Rules instance = rules0
      List of interfaces monitored by this virtual sensor = mypair
      Denied Address Information
         Number of Active Denied Attackers = 2
         Number of Denied Attackers Inserted = 0
         Number of Denied Attackers Total Hits = 0
         Number of times max-denied-attackers limited creation of new entry = 0
         Number of exec Clear commands during uptime = 1
      Denied Attackers and hit count for each.
         10.20.2.5 = 0
         10.20.5.2 = 0
```

The statistics have all been cleared except for the `Number of Active Denied Attackers` and `Number of exec Clear commands during uptime` categories. It is important to know if the list has been cleared.

Event Action Rules Example

The following example demonstrates how the individual components of your event action rules work together.

Risk Rating Ranges for Example 1

- **Produce Alert**—1-100
- **Produce Verbose Alert**—90-100
- **Request SNMP Trap**—50-100
- **Log Pair Packets**—90-100
- **Log Victim Packets**—90-100
- **Log Attacker Packets**—90-100
- **Reset TCP Connection**—90-100
- **Request Block Connection**—70-89
- **Request Block Host**—90-100
- **Deny Attacker Inline**—0-0
- **Deny Connection Inline**—90-100
- **Deny Packet Inline**—90-100

Event Action Filters for Example 1

1. SigID=2004, Attacker Address=*, Victim Address=20.1.1.1, Actions to Remove=ALL, Risk Rating Range=1-100, StopOnMatch=True
2. SigID=2004, Attacker Address=30.1.1.1, Victim Address=*, Actions to Remove=ALL, Risk Rating Range=1-100, StopOnMatch=True
3. SigID=2004, Attacker Address=*, Victim Address=*, Actions to Remove=None, Risk Rating Range=95-100, StopOnMatch=True
4. SigID=2004, Attacker Address=*, Victim Address=*, Actions to Remove=denyAttackerInline, requestBlockHost, requestBlockConnection, Risk Rating Range=56-94, StopOnMatch=True
5. SigID=2004, Attacker Address=*, Victim Address=*, Actions to Remove=denyAttackerInline, requestBlockHost, produceAlert, resetTcpConnection, logAttackerPackets, Risk Rating Range=1-55, StopOnMatch=True

Results for Example 1

When SIG 2004 is detected:

- If the attacker address is 30.1.1.1 or the victim address is 20.1.1.1, the event is consumed (ALL actions are subtracted).

If the attacker address is not 30.1.1.1 and the victim address is not 20.1.1.1:

- If the RR is 50, Produce Alert and Request SNMP Trap are added by the event action override component, but Produce Alert is subtracted by the event action filter. However, the event action policy forces the alert action because Request SNMP Trap is dependent on the <evIdsAlert>.
- If the RR is 89, Request SNMP Trap and Request Block Connection are added by the event action override component. However, Request Block Connection is subtracted by the event action filter.
- If the RR is 96, all actions except Deny Attacker Inline and Request Block Connection are added by the event action override component, and none are removed by the event action filter. The third filter line with the filter action NONE is optional, but is presented as a clearer way to define this type of filter.

Monitoring Events

This section describes how to display and clear events from the Event Store, and contains the following topics:

- Displaying Events, page 6-24
- Clearing Events from Event Store, page 6-27

Displaying Events

Use the **show events** [{[**alert** [informational] [low] [medium] [high] [**include-traits** ***traits***] [**exclude-traits** ***traits***]] | **error** [warning] [error] [fatal] | **NAC** | **status**}] [*hh:mm:ss* [*month day* [*year*]] | **past** *hh:mm:ss*] command to display events from the Event Store.

Events are displayed beginning at the start time. If you do not specify a start time, events are displayed beginning at the current time. If you do not specify an event type, all events are displayed.

Note Events are displayed as a live feed until you cancel the request by pressing Ctrl-C.

The following options apply:

- **alert**—Displays alerts. Provides notification of some suspicious activity that may indicate an attack is in process or has been attempted.

  If no level is selected (informational, low, medium, or high), all alert events are displayed.
- **include-traits**—Displays alerts that have the specified traits.
- **exclude-traits**—Does not display alerts that have the specified traits.
- **traits**—Trait bit position in decimal (0 to 15).
- **error**—Displays error events. Error events are generated by services when error conditions are encountered.

- **NAC**—Displays Attack Response Controller (ARC) requests.

  Note ARC is formerly known as Network Access Controller (NAC). This name change has not been completely implemented throughout the IDM and CLI for IPS 5.1.

- **status**—Displays status events.
- **past**—Displays events starting in the past for the specified hours, minutes, and seconds.
- *hh:mm:ss*—Hours, minutes, and seconds in the past to begin the display.

Note The **show events** command waits until a specified event is available. It continues to wait and display events until you exit by pressing Ctrl-C.

To display events from the Event Store, follow these steps:

Step 1 Log in to the CLI.

Step 2 Display all events starting now:

```
sensor#@ show events
evError: eventId=1041472274774840147 severity=warning vendor=Cisco
  originator:
    hostId: sensor2
    appName: cidwebserver
    appInstanceId: 12075
  time: 2003/01/07 04:41:45 2003/01/07 04:41:45 UTC
  errorMessage: name=errWarning received fatal alert: certificate_unknown

evError: eventId=1041472274774840148 severity=error vendor=Cisco
  originator:
    hostId: sensor2
    appName: cidwebserver
    appInstanceId: 351
  time: 2003/01/07 04:41:45 2003/01/07 04:41:45 UTC
  errorMessage: name=errTransport WebSession::sessionTask(6) TLS connection exce
ption: handshake incomplete.
```

The feed continues showing all events until you press **Ctrl-C**.

Step 3 Display the block requests beginning at 10:00 a.m. on February 9, 2005:

```
sensor#@ show events NAC 10:00:00 Feb 9 2005
evShunRqst: eventId=1106837332219222281 vendor=Cisco
  originator:
    deviceName: Sensor1
    appName: NetworkAccessControllerApp
    appInstance: 654
  time: 2005/02/09 10:33:31 2004/08/09 13:13:31
  shunInfo:
    host: connectionShun=false
      srcAddr: 11.0.0.1
      destAddr:
      srcPort:
      destPort:
      protocol: numericType=0 other
    timeoutMinutes: 40
  evAlertRef: hostId=esendHost 123456789012345678
sensor#
```

Step 4 Display errors with the warning level starting at 10:00 a.m. February 9 2005:

```
sensor# show events error warning 10:00:00 Feb 9 2005
evError: eventId=1041472274774840197 severity=warning vendor=Cisco
  originator:
    hostId: sensor
    appName: cidwebserver
    appInstanceId: 12160
  time: 2003/01/07 04:49:25 2003/01/07 04:49:25 UTC
  errorMessage: name=errWarning received fatal alert: certificate_unknown
```

Step 5 Display alerts from the past 45 seconds:

```
sensor# show events alert past 00:00:45

evIdsAlert: eventId=1109695939102805307 severity=medium vendor=Cisco
  originator:
    hostId: sensor
    appName: sensorApp
    appInstanceId: 367
  time: 2005/03/02 14:15:59 2005/03/02 14:15:59 UTC
  signature: description=Nachi Worm ICMP Echo Request id=2156 version=S54
    subsigId: 0
    sigDetails: Nachi ICMP
  interfaceGroup:
  vlan: 0
  participants:
    attacker:
      addr: locality=OUT 10.89.228.202
    target:
      addr: locality=OUT 10.89.150.185
  riskRatingValue: 70
  interface: fe0_1
  protocol: icmp


evIdsAlert: eventId=1109695939102805308 severity=medium vendor=Cisco
  originator:
--MORE--
```

Step 6 Display events that began 30 seconds in the past:

```
sensor# show events past 00:00:30
evStatus: eventId=1041526834774829055 vendor=Cisco
  originator:
    hostId: sensor
    appName: mainApp
    appInstanceId: 2215
  time: 2003/01/08 02:41:00 2003/01/08 02:41:00 UTC
  controlTransaction: command=getVersion successful=true
    description: Control transaction response.
    requestor:
      user: cids
      application:
        hostId: 64.101.182.101
        appName: -cidcli
        appInstanceId: 2316


evStatus: eventId=1041526834774829056 vendor=Cisco
  originator:
    hostId: sensor
    appName: login(pam_unix)
    appInstanceId: 2315
  time: 2003/01/08 02:41:00 2003/01/08 02:41:00 UTC
```

```
  syslogMessage:
    description: session opened for user cisco by cisco(uid=0)
```

Clearing Events from Event Store

Use the **clear events** command to clear Event Store.

To clear events from Event Store, follow these steps:

Step 1 Log in to the CLI using an account with administrator privileges.

Step 2 Clear Event Store:

```
sensor# clear events
Warning: Executing this command will remove all events currently stored in the event
store.
Continue with clear? []:
```

Step 3 Enter **yes** to clear the events.

CHAPTER 7

Defining Signatures

This chapter describes how to define and create signatures. It contains the following sections:

Understanding Signatures

Attacks or other misuses of network resources can be defined as network intrusions. Sensors that use a signature-based technology can detect network intrusions. A *signature* is a set of rules that your sensor uses to detect typical intrusive activity, such as DoS attacks. As sensors scan network packets, they use signatures to detect known attacks and respond with actions that you define.

The sensor compares the list of signatures with network activity. When a match is found, the sensor takes an action, such as logging the event or sending an alert. Sensors let you modify existing signatures and define new ones.

Signature-based intrusion detection can produce false positives because certain normal network activity can be misinterpreted as malicious activity. For example, some network applications or operating systems may send out numerous ICMP messages, which a signature-based detection system might interpret as an attempt by an attacker to map out a network segment. You can minimize false positives by tuning your signatures.

To configure a sensor to monitor network traffic for a particular signature, you must enable the signature. By default, the most critical signatures are enabled when you install the signature update. When an attack is detected that matches an enabled signature, the sensor generates an alert, which is stored in the sensor's event store. The alerts, as well as other events, may be retrieved from the event store by web-based clients. By default the sensor logs all Informational alerts or higher.

Some signatures have subsignatures, that is, the signature is divided into subcategories. When you configure a subsignature, changes made to the parameters of one subsignature apply only to that subsignature. For example, if you edit signature 3050 subsignature 1 and change the severity, the severity change applies to only subsignature 1 and not to 3050 2, 3050 3, and 3050 4.

IPS 5.1 contains over 1000 built-in default signatures. You cannot rename or delete signatures from the list of built-in signatures, but you can retire signatures to remove them from the sensing engine. You can later activate retired signatures; however, this process requires the sensing engines to rebuild their

configuration, which takes time and could delay the processing of traffic. You can tune built-in signatures by adjusting several signature parameters. Built-in signatures that have been modified are called *tuned* signatures.

You can create signatures, which are called *custom* signatures. Custom signature IDs begin at 60000. You can configure them for several things, such as matching of strings on UDP connections, tracking of network floods, and scans. Each signature is created using a signature engine specifically designed for the type of traffic being monitored.

Signature Variables

This section describes signature variables, and contains the following topics:

- Understanding Signature Variables, page 7-2
- Configuring Signature Variables, page 7-2

Understanding Signature Variables

When you want to use the same value within multiple signatures, use a variable. When you change the value of a variable, the variables in all signatures are updated. This saves you from having to change the variable repeatedly as you configure signatures.

Note You must preface the variable with a dollar ($) sign to indicate that you are using a variable rather than a string.

Some variables cannot be deleted because they are necessary to the signature system. If a variable is protected, you cannot select it to edit it. You receive an error message if you try to delete protected variables. You can edit only one variable at a time.

Configuring Signature Variables

Use the **variables** command in the signature definition submode to create variables.

The following options apply:

- *variable-name*—Identifies the name assigned to this variable.

  A valid name can only contain numbers or letters. You can also use a hyphen (-) or underscore (_).
- **ip-addr-range**—System-defined variable for grouping IP addresses.

  The valid values are: A.B.C.D-A.B.C.D[,A.B.C.D-A.B.C.D]
- **web-ports**—System-defined variable for ports to look for HTTP traffic.

  To designate multiple port numbers for a single variable, place a comma between the entries. For example, 80, 3128, 8000, 8010, 8080, 8888, 24326.

To configure signature variables, follow these steps:

Step 1 Log in to the CLI using an account with administrator or operator privileges.

Step 2 Enter signature definition submode:

```
sensor# configure terminal
sensor(config)# service signature-definition sig0
```

Step 3 Create a signature variable for a group of IP addresses:

```
sensor(config-sig)# variables IPADD ip-addr-range 10.1.1.1-10.1.1.24
```

Step 4 Edit the signature variable for web ports:

```
sensor(config-sig)# variables WEBPORTS web-ports 80,3128,8000
```

WEBPORTS has a predefined set of ports where web servers are running, but you can edit the value. This variable affects all signatures that have web ports. The default is 80, 3128, 8000, 8010, 8080, 8888, 24326.

Step 5 Verify the changes:

```
sensor(config-sig)# show settings
   variables (min: 0, max: 256, current: 2)
   ----------------------------------------------
      variable-name: IPADD
      ----------------------------------------------
         ip-addr-range: 10.1.1.1-10.1.1.24
      ----------------------------------------------
      <protected entry>
      variable-name: WEBPORTS
      ----------------------------------------------
         web-ports: 80,3128,8000 default: 80-80,3128-3128,8000-8000,8010-8010,80
80-8080,8888-8888,24326-24326
      ----------------------------------------------
```

Step 6 Exit signature definition submode:

```
sensor(config-sig)# exit
Apply Changes:?[yes]:
```

Step 7 Press **Enter** to apply the changes or enter **no** to discard them.

Configuring Signatures

This section describes how to configure signature parameters, and contains the following topics:

Configuring General Signature Parameters

The following options apply to configuring the general parameters of a specific signature:

- **alert-frequency**—Sets the summary options for grouping alerts.

  For the procedure, see Configuring Alert Frequency, page 7-5.
- **alert-severity**—Sets the severity of the alert.

  For the procedure, see Configuring Alert Severity, page 7-6.
- **engine**—Specifies the signature engine. You can assign actions when you are in the engine submode.

  For more information about signature engines, see Appendix B, "Signature Engines." For the procedure for assigning actions, see Assigning Actions to Signatures, page 7-11.
- **event-counter**—Sets the event count.

  For the procedure, see Configuring Event Counter, page 7-8.
- **promisc-delta**—The delta value used to determine the seriousness of the alert.

Caution We do not recommend that you change the promisc-delta setting for a signature.

  Promiscuous delta lowers the RR of certain alerts in promiscuous mode. Because the sensor does not know the attributes of the target system and in promiscuous mode cannot deny packets, it is useful to lower the prioritization of promiscuous alerts (based on the lower RR) so the administrator can focus on investigating higher RR alerts.

  In inline mode, the sensor can deny the offending packets and they never reach the target host, so it does not matter if the target was vulnerable. The attack was not allowed on the network and so we do not subtract from the RR value.

  Signatures that are not service, OS, or application specific have 0 for the promiscuously delta. If the signature is specific to an OS, service, or application, it has a promiscuous delta of 5, 10, or 15 calculated from 5 points for each category.
- **sig-description**—Your description of the signature.
- **sig-fidelity-rating**—Rating of the fidelity of signature.

  For the procedure, see Configuring Signature Fidelity Rating, page 7-9.
- **status**—Sets the status of the signature to enabled or retired.

  For the procedure, see Configuring the Status of Signatures, page 7-10.

Configuring Alert Frequency

Use the **alert-frequency** command in the signature definition submode to configure the alert frequency for a signature.

The following options apply:

- *sig_id*—Identifies the unique numerical value assigned to this signature.

  This value lets the sensor identify a particular signature. The value is 1000 to 65000.
- *subsig_id*—Identifies the unique numerical value assigned to this subsignature.

  A subsignature ID is used to identify a more granular version of a broad signature. The value is 0 to 255.
- **alert-frequency**—How often the sensor alerts you when this signature is firing.

  Specify the following parameters for this signature:
  - **summary-mode**—The way you want the sensor to group the alerts:

    fire-all—Fires an alert on all events.

    fire-once—Fires an alert only once.

    global-summarize—Summarizes an alert so that it only fires once regardless of how many attackers or victims.

    summarize—Summarize all the alerts.
  - **summary-interval**—Time in seconds used in each summary alert.

    The value is 1 to 65535.
  - **summary-key**—Storage type on which to summarize this signature.

    Axxx—Attacker address.

    Axxb—Attacker address and victim port.

    AxBx—Attacker and victim addresses.

    AaBb—Attacker and victim addresses and ports.

    xxBx—Victim address.
  - **specify-global-summary-threshold {yes | no}**—Specifies whether you want to configure a global summary threshold (optional).
  - **global-summary-threshold**—Threshold number of events to take alert into global summary.

To configure the alert frequency parameters of a signature, follow these steps:

Step 1 Log in to the CLI using an account with administrator or operator privileges.

Step 2 Enter signature definition submode:

```
sensor# configure terminal
sensor(config)# service signature-definition sig0
```

Step 3 Specify the signature you want to configure:

```
sensor(config-sig)# signatures 9000 0
```

Step 4 Enter alert frequency submode:

```
sensor(config-sig-sig)# alert-frequency
```

Step 5 Configure the alert frequency of this signature:

a. Configure the summary mode to, for example, fire once:

```
sensor(config-sig-sig-ale)# summary-mode fire-once
sensor(config-sig-sig-ale-fir)# specify-global-summary-threshold yes
sensor(config-sig-sig-ale-fir-yes)# global-summary-threshold 3000
sensor(config-sig-sig-ale-fir-yes)# summary-interval 5000
```

b. Configure the summary key:

```
sensor(config-sig-sig-ale-fir-yes)# exit
sensor(config-sig-sig-ale-fir)# summary-key AxBx
```

c. Verify the settings:

```
sensor(config-sig-sig-ale-fir)# show settings
   fire-once
   ----------------------------------------------
      summary-key: AxBx default: Axxx
      specify-global-summary-threshold
      ---------------------------------------------
         yes
         ---------------------------------------------
            global-summary-threshold: 3000 default: 120
            summary-interval: 5000 default: 15
         ---------------------------------------------
      ---------------------------------------------
   ----------------------------------------------
sensor(config-sig-sig-ale-fir)#
```

Step 6 Exit alert-frequency submode:

```
sensor(config-sig-sig-ale-fir)# exit
sensor(config-sig-sig-ale)# exit
sensor(config-sig-sig)# exit
sensor(config-sig)# exit
Apply Changes:?[yes]:
```

Step 7 Press **Enter** to apply the changes or enter **no** to discard them.

Configuring Alert Severity

Use the **alert-severity** command in the signature definition submode to configure the severity of a signature.

The following options apply:

- *sig_id*—Identifies the unique numerical value assigned to this signature.

  This value lets the sensor identify a particular signature. The value is 1000 to 65000.
- *subsig_id*—Identifies the unique numerical value assigned to this subsignature.

  A subsignature ID is used to identify a more granular version of a broad signature. The value is 0 to 255.
- **alert-severity**—Severity of the alert:
  - **high** —Dangerous alert.
  - **medium**—Medium level alert.

- **low**—Low level alert.
- **informational**—Informational alert.

  This is the default.

To configure the alert severity, follow these steps:

Step 1 Log in to the CLI using an account with administrator or operator privileges.

Step 2 Enter signature definition submode:

```
sensor# configure terminal
sensor(config)# service signature-definition sig0
```

Step 3 Choose the signature you want to configure:

```
sensor(config-sig)# signatures 9000 0
```

Step 4 Assign the alert severity:

```
sensor(config-sig-sig)# alert-severity medium
```

Step 5 Verify the settings:

```
sensor(config-sig-sig)# show settings
   <protected entry>
   sig-id: 9000
   subsig-id: 0
   ----------------------------------------------
      alert-severity: medium default: informational
      sig-fidelity-rating: 75 <defaulted>
      promisc-delta: 0 <defaulted>
      sig-description
      ----------------------------------------------
         sig-name: Back Door Probe (TCP 12345) <defaulted>
         sig-string-info: SYN to TCP 12345 <defaulted>
         sig-comment: <defaulted>
         alert-traits: 0 <defaulted>
         release: 40 <defaulted>
      ----------------------------------------------
      engine
      ----------------------------------------------
         atomic-ip
         ----------------------------------------------
            event-action: produce-alert <defaulted>
            fragment-status: any <defaulted>
            specify-l4-protocol
            ----------------------------------------------
--MORE--
```

Step 6 Exit signatures submode:

```
sensor(config-sig-sig)# exit
sensor(config-sig)# exit
Apply Changes:?[yes]:
```

Step 7 Press **Enter** to apply the changes or enter **no** to discard them.

Configuring Event Counter

Use the **event-counter** command in the signature definition submode to configure how the sensor counts events. For example, you can specify that you want the sensor to send an alert only if the same signature fires 5 times for the same address set.

The following options apply:

- **event-count**—The number of times an event must occur before an alert is generated. The valid range is 1 to 65535. The default is 1.
- **event-count-key**—The storage type on which to count events for this signatures.
  - **Axxx**—Attacker address
  - **AxBx**—Attacker and victim addresses
  - **Axxb**—Attacker address and victim port
  - **xxBx**—Victim address
  - **AaBb**—Attacker and victim addresses and ports
- **specify-alert-interval {yes | no}**—Enables alert interval.
  - **alert-interval**—The time in seconds before the event count is reset. The default is 60.

To configure event counter, follow these steps:

Step 1 Log in to the CLI using an account with administrator or operator privileges.

Step 2 Enter signature definition submode:

```
sensor# configure terminal
sensor(config)# service signature-definition sig0
```

Step 3 Choose the signature for which you want to configure event counter:

```
sensor(config-sig)# signatures 9000 0
```

Step 4 Enter event counter submode:

```
sensor(config-sig-sig)# event-counter
```

Step 5 Configure how many times an event must occur before an alert is generated:

```
sensor(config-sig-sig-eve)# event-count 2
```

Step 6 Configure the storage type on which you want to count events for this signature:

```
sensor(config-sig-sig-eve)# event-count-key AxBx
```

Step 7 (Optional) Enable alert interval:

```
sensor(config-sig-sig-eve)# specify-alert-interval yes
```

Step 8 (Optional) Specify the amount of time in seconds before the event count should be reset:

```
sensor(config-sig-sig-eve-yes)# alert-interval 30
```

Step 9 Verify the settings:

```
sensor(config-sig-sig-eve-yes)# exit
sensor(config-sig-sig-eve)# show settings
   event-counter
   ----------------------------------------------
      event-count: 2 default: 1
```

```
      event-count-key: AxBx default: Axxx
      specify-alert-interval
      ----------------------------------------------
         yes
         ---------------------------------------------
            alert-interval: 30 default: 60
         ---------------------------------------------
      ----------------------------------------------
   ---------------------------------------------
sensor(config-sig-sig-eve)#
```

Step 10 Exit signatures submode:

```
sensor(config-sig-sig-eve)# exit
sensor(config-sig-sig)# exit
sensor(config-sig)# exit
Apply Changes:?[yes]:
```

Step 11 Press **Enter** to apply the changes or enter **no** to discard them.

Configuring Signature Fidelity Rating

Use the **sig-fidelity-rating** command in the signature definition submode to configure the signature fidelity rating for a signature.

The following option applies:

- **sig-fidelity-rating**—Identifies the weight associated with how well this signature might perform in the absence of specific knowledge of the target.

  The valid value is 0 to 100.

To configure the signature fidelity rating for a signature, follow these steps:

Step 1 Log in to the CLI using an account with administrator or operator privileges.

Step 2 Enter signature definition submode:

```
sensor# configure terminal
sensor(config)# service signature-definition sig0
```

Step 3 Choose the signature you want to configure:

```
sensor(config-sig)# signatures 12000 0
```

Step 4 Configure the fidelity rating for this signature:

```
sensor(config-sig-sig)# sig-fidelity-rating 50
```

Step 5 Verify the settings:

```
sensor(config-sig-sig)# show settings
   <protected entry>
   sig-id: 12000
   subsig-id: 0
   ----------------------------------------------
      alert-severity: low <defaulted>
      sig-fidelity-rating: 50 default: 85
      promisc-delta: 15 <defaulted>
      sig-description
      ----------------------------------------------
```

```
        sig-name: Gator Spyware Beacon <defaulted>
        sig-string-info: /download/ User-Agent: Gator <defaulted>
        sig-comment: <defaulted>
        alert-traits: 0 <defaulted>
        release: 71 <defaulted>
    ----------------------------------------------
```

Step 6 Exit signatures submode:

```
sensor(config-sig-sig)# exit
sensor(config-sig)# exit
Apply Changes:?[yes]:
```

Step 7 Press **Enter** to apply the changes or enter **no** to discard them.

Configuring the Status of Signatures

Use the **status** command in the signature definition submode to specify the status of a specific signature. The following options apply:

- **status**—Identifies whether the signature is enabled, disabled, or retired.
  - **enabled {true | false}**—Enables the signature.
  - **retired {true | false}**—Retires the signature.

Caution Activating and retiring signatures can take 30 minutes or longer.

To change the status of a signature, follow these steps:

Step 1 Log in to the CLI using an account with administrator or operator privileges.

Step 2 Enter signature definition submode:

```
sensor# configure terminal
sensor(config)# service signature-definition sig0
```

Step 3 Choose the signature you want to configure:

```
sensor(config-sig)# signatures 12000 0
```

Step 4 Change the status for this signature:

```
sensor(config-sig-sig)# status
sensor(config-sig-sig-sta)# enabled true
```

Step 5 Verify the settings:

```
sensor(config-sig-sig-sta)# show settings
   status
   ----------------------------------------------
      enabled: true default: false
      retired: false <defaulted>
   ----------------------------------------------
sensor(config-sig-sig-sta)#
```

Step 6 Exit signatures submode:

```
sensor(config-sig-sig-sta)# exit
```

```
sensor(config-sig-sig)# exit
sensor(config-sig)# exit
Apply Changes:?[yes]:
```

Step 7 Press **Enter** to apply the changes or enter **no** to discard them.

Assigning Actions to Signatures

Use the **event-action** command in the signature definition submode to configure the actions the sensor takes when the signature fires.

The following options apply:

- **deny-attacker-inline** —(Inline mode only) Does not transmit this packet and future packets from the attacker address for a specified period of time.
- **deny-attacker-service-pair-inline**—(Inline only) Does not transmit this packet and future packets on the attacker address victim port pair for a specified period of time.
- **deny-attacker-victim-pair-inline**—(Inline only) Does not transmit this packet and future packets on the attacker/victim address pair for a specified period of time.
- **deny-connection-inline**—(Inline mode only) Does not transmit this packet and future packets on the TCP Flow.
- **deny-packet-inline**—(Inline mode only) Does not transmit this packet.
- **log-attacker-packets**—Starts IP logging of packets containing the attacker address. This action causes an alert to be written to the Event Store, even if **produce-alert** is not selected.
- **log-pair-packets**—Starts IP logging of packets containing the attacker-victim address pair. This action causes an alert to be written to the Event Store, even if **produce-alert** is not selected.
- **log-victim-packets**—Starts IP logging of packets containing the victim address. This action causes an alert to be written to the Event Store, even if **produce-alert** is not selected.
- **produce-alert** —Writes the event to the Event Store as an alert.
- **produce-verbose-alert**—Includes an encoded dump (possibly truncated) of the offending packet in the alert. This action causes an alert to be written to the Event Store, even if **produce-alert** is not selected.
- **request-block-connection**—Sends a request to ARC to block this connection. You must have blocking devices configured to implement this action.
- **request-block-host**—Sends a request to ARC to block this attacker host. You must have blocking devices configured to implement this action.
- **request-rate-limit**—Sends a rate limit request to ARC to perform rate limiting. You must have rate limiting devices configured to implement this action.
- **request-snmp-trap**—Sends a request to the Notification Application component of the sensor to perform SNMP notification. This action causes an alert to be written to the Event Store, even if **produce-alert** is not selected. You must have SNMP configured on the sensor to implement this action.

- **reset-tcp-connection**—Sends TCP resets to hijack and terminate the TCP flow. **Reset TCP Connection** only works on TCP signatures that analyze a single connection. It does not work for sweeps or floods.
- **modify-packet-inline**— Modifies packet data to remove ambiguity about what the end point might do with the packet.

Understaning Deny Packet Inline

For signatures that have deny-packet-inline configured as an action or for an event action override that adds deny-packet-inline as an action, the following actions may be taken:

- droppedPacket
- deniedFlow
- tcpOneWayResetSent

The deny packet inline action is represented as a dropped packet action in the alert. When a deny packet inline occurs for a TCP connection, it is automatically upgraded to a deny connection inline action and seen as a denied flow in the alert. If the IPS denies just one packet, the TCP continues to try to send that same packet again and again, so the IPS denies the entire connection to ensure it never succeeds with the resends.

When a deny connection inline occurs, the IPS also automatically sends a TCP one-way reset, which shows up as a TCP one-way reset sent in the alert. When the IPS denies the connection, it leaves an open connection on both the client (generally the attacker) and the server (generally the victim). Too many open connections can result in resource problems on the victim. So the IPS sends a TCP reset to the victim to close the connection on the victim side (usually the server), which conserves the resources of the victim. It also prevents a failover that would otherwise allow the connection to fail over to a different network path and reach the victim. The IPS leaves the attacker side open and denies all traffic from it.

To configure event actions for a signature, follow these steps:

Step 1 Log in to the CLI using an account with administrator privileges.

Step 2 Enter signature definition mode:

```
sensor# configure terminal
sensor(config)# service signature-definition sig0
sensor(config-sig)#
```

Step 3 Choose the signature you want to configure:

```
sensor(config-sig)# signatures 1200 0
```

Step 4 Enter the normalizer engine:

```
sensor(config-sig-sig)# engine normalizer
```

Step 5 Configure the event action:

```
sensor(config-sig-sig-nor)# event-action produce-alert|request-snmp-trap
```

Note Each time you configure the event actions for a signature, you overwrite the previous configuration. For example, if you always want to produce an alert when the signature is fired, you must configure it along with the other event actions you want. Use the | symbol to add more than one event action, for example, **product-alert|deny-packet-inline|request-snmp-trap**.

Step 6 Verify the settings:

```
sensor(config-sig-sig-nor)# show settings
   normalizer
   ---------------------------------------------
      event-action: produce-alert|request-snmp-trap default:
produce-alert|deny-packet-inline
```

Step 7 Exit event action submode:

```
sensor(config-sig-sig-nor)# exit
sensor(config-sig-sig)# exit
sensor(config-sig)# exit
Apply Changes:?[yes]:
```

Step 8 Press **Enter** to apply the changes or enter **no** to discard them.

Configuring AIC Signatures

This section describes the AIC signatures and how to configure them. It contains the following topics:

Overview

AIC provides detailed analysis of web traffic. It provides granular control over HTTP sessions to prevent abuse of the HTTP protocol. It also allows administrative control over applications that attempt to tunnel over specified ports, such as instant messaging, and tunneling applications such as, gotomypc. Inspection and policy checks for P2P and instant messaging is possible if these applications are running over HTTP.

AIC also provides a way to inspect FTP traffic and control the commands being issued. You can enable or disable the predefined signatures or you can create policies through custom signatures.

The AIC engine runs when HTTP traffic is received on AIC web ports. If traffic is web traffic, but not received on the AIC web ports, the Service HTTP engine is executed. AIC inspection can be on any port if it is configured as an AIC web port and the traffic to be inspected is HTTP traffic.

Caution The AIC web ports are regular HTTP web ports. You can turn on AIC web ports to distinguish which ports should watch for regular HTTP traffic and which ports should watch for AIC enforcement. You might use AIC web ports, for example, if you have a proxy on port 82 and you need to monitor it. We recommend that you do not configure separate ports for AIC enforcement.

AIC has the following categories of signatures:

- HTTP request method
  - Define request method
  - Recognized request methods

  For a list of signature IDs and descriptions, see AIC Request Method Signatures, page 7-16.
- MIME type
  - Define content type
  - Recognized content type

  For a list of signature IDs and descriptions, see AIC MIME Define Content Type Signatures, page 7-17. For the procedure for creating a custom MIME signature, see Example AIC MIME-Type Signature, page 7-42.
- Define web traffic policy

  There is one predefined signature, 12674, that specifies the action to take when noncompliant HTTP traffic is seen. The parameter Alarm on Non HTTP Traffic enables the signature. By default this signature is enabled.
- Transfer encodings
  - Associate an action with each method
  - List methods recognized by the sensor
  - Specify which actions need to be taken when a chunked encoding error is seen

  For a list of signature IDs and descriptions, see AIC Transfer Encoding Signatures, page 7-20.
- FTP commands

  Associates an action with an FTP command. For a list of signature IDs and descriptions, see AIC FTP Commands Signatures, page 7-20.

Configuring the Application Policy

Use the **application-policy** command in the signature definition submode to enable the web AIC feature. You can configure the sensor to provide Layer 4 to Layer 7 packet inspection to prevent malicious attacks related to web and FTP services.

The following options apply:

- **ftp-enable {true | false}**—Enables protection for FTP services. Set to true to require the sensor to inspect FTP traffic.

  The default is false.
- **http-policy**—Enables inspection of HTTP traffic.
  - **aic-web-ports**—Variable for ports to look for AIC traffic.

    The valid range is 0 to 65535. A comma-separated list of integer ranges a-b[,c-d] within 0-65535. The second number in the range must be greater than or equal to the first number.

    The default is 80-80,3128-3128,8000-8000,8010-8010,8080-8080,8888-8888,24326-24326.

    Note We recommend that you not configure AIC web ports, but rather use the default web ports.

- **http-enable {true | false}**—Enables protection for web services. Set to true to require the sensor to inspect HTTP traffic for compliance with the RFC.

  The default is false.

- **max-outstanding-http-requests-per-connection**—Maximum allowed HTTP requests per connection.

  The valid value is 1 to 16. The default is 10.

To configure the application policy, follow these steps:

Step 1 Log in to the CLI using an account with administrator or operator privileges.

Step 2 Enter application policy submode:

```
sensor# configure terminal
sensor(config)# service signature-definition sig0
sensor(config-sig)# application-policy
```

Step 3 Enable inspection of FTP traffic:

```
sensor(config-sig-app)# ftp-enable true
```

Step 4 Configure the HTTP application policy:

a. Enter HTTP application policy submode:

```
sensor(config-sig-app)# http-policy
```

b. Enable HTTP application policy enforcement:

```
sensor(config-sig-app-htt)# http-enable true
```

c. Specify the number of outstanding HTTP requests per connection that can be outstanding without having received a response from the server:

```
sensor(config-sig-app-htt)# max-outstanding-http-requests-per-connection 5
```

d. (Optional) Edit the AIC ports:

```
sensor(config-sig-app-htt)# aic-web-ports 80-80,3128-3128
```

Note We recommend that you not configure AIC web ports, but rather use the default web ports.

Step 5 Verify your settings:

```
sensor(config-sig-app)# show settings
   application-policy
   ----------------------------------------------
      http-policy
      ---------------------------------------------
         http-enable: true default: false
         max-outstanding-http-requests-per-connection: 5 default: 10
         aic-web-ports: 80-80,3128-3128 default: 80-80,3128-3128,8000-8000,8010-
8010,8080-8080,8888-8888,24326-24326
      ---------------------------------------------
      ftp-enable: true default: false
   ----------------------------------------------
sensor(config-sig-app)#
```

Step 6 Exit signature definition submode:

```
sensor(config-sig-app)# exit
```

```
sensor(config-sig)# exit
Apply Changes:?[yes]:
```

Step 7 Press **Enter** to apply the changes or enter **no** to discard them.

AIC Request Method Signatures

The HTTP request method has two categories of signatures:

- Define request method—Allows actions to be associated with request methods. You can expand and modify the signatures (Define Request Method).
- Recognized request methods—Lists methods that are recognized by the sensor (Recognized Request Methods).

Table 7-1 lists the predefined define request method signatures. Enable the signatures that have the predefined method you need. For the procedure for enabling signatures, see Configuring the Status of Signatures, page 7-10.

Table 7-1 Request Method Signatures

Signature ID	Define Request Method
12676	Request Method Not Recognized
12677	Define Request Method PUT
12678	Define Request Method CONNECT
12679	Define Request Method DELETE
12680	Define Request Method GET
12681	Define Request Method HEAD
12682	Define Request Method OPTIONS
12683	Define Request Method POST
12685	Define Request Method TRACE
12695	Define Request Method INDEX
12696	Define Request Method MOVE
12697	Define Request Method MKDIR
12698	Define Request Method COPY
12699	Define Request Method EDIT
12700	Define Request Method UNEDIT
12701	Define Request Method SAVE
12702	Define Request Method LOCK
12703	Define Request Method UNLOCK
12704	Define Request Method REVLABEL
12705	Define Request Method REVLOG
12706	Define Request Method REVADD
12707	Define Request Method REVNUM

Table 7-1 Request Method Signatures (continued)

Signature ID	Define Request Method
12708	Define Request Method SETATTRIBUTE
12709	Define Request Method GETATTRIBUTENAME
12710	Define Request Method GETPROPERTIES
12711	Define Request Method STARTENV
12712	Define Request Method STOPREV

AIC MIME Define Content Type Signatures

There are two policies associated with MIME types:

- Define content type—Associates specific actions for the following cases (Define Content Type):
  - Deny a specific MIME type, such as an image/jpeg
  - Message size violation
  - MIME-type mentioned in header and body do not match
- Recognized content type (Recognized Content Type)

Table 7-2 lists the predefined define content type signatures. Enable the signatures that have the predefined content type you need. For the procedure for enabling signatures, see Configuring the Status of Signatures, page 7-10. You can also create custom define content type signatures. For the procedure, see Example AIC MIME-Type Signature, page 7-42.

Table 7-2 Define Content Type Signatures

Signature ID	Signature Description
12621	Content Type image/gif Invalid Message Length
12622 2	Content Type image/png Verification Failed
12623 0 12623 1 12623 2	Content Type image/tiff Header Check Content Type image/tiff Invalid Message Length Content Type image/tiff Verification Failed
12624 0 12624 1 12624 2	Content Type image/x-3ds Header Check Content Type image/x-3ds Invalid Message Length Content Type image/x-3ds Verification Failed
12626 0 12626 1 12626 2	Content Type image/x-portable-bitmap Header Check Content Type image/x-portable-bitmap Invalid Message Length Content Type image/x-portable-bitmap Verification Failed
12627 0 12627 1 12627 2	Content Type image/x-portable-graymap Header Check Content Type image/x-portable-graymap Invalid Message Length Content Type image/x-portable-graymap Verification Failed
12628 0 12628 1 12628 2	Content Type image/jpeg Header Check Content Type image/jpeg Invalid Message Length Content Type image/jpeg Verification Failed
12629 0 12629 1	Content Type image/cgf Header Check Content Type image/cgf Invalid Message Length

Table 7-2 Define Content Type Signatures (continued)

Signature ID	Signature Description
12631 0 12631 1	Content Type image/x-xpm Header Check Content Type image/x-xpm Invalid Message Length
12633 0 12633 1 12633 2	Content Type audio/midi Header Check Content Type audio/midi Invalid Message Length Content Type audio/midi Verification Failed
12634 0 12634 1 12634 2	Content Type audio/basic Header Check Content Type audio/basic Invalid Message Length Content Type audio/basic Verification Failed
12635 0 12635 1 12635 2	Content Type audio/mpeg Header Check Content Type audio/mpeg Invalid Message Length Content Type audio/mpeg Verification Failed
12636 0 12636 1 12636 2	Content Type audio/x-adpcm Header Check Content Type audio/x-adpcm Invalid Message Length Content Type audio/x-adpcm Verification Failed
12637 0 12637 1 12637 2	Content Type audio/x-aiff Header Check Content Type audio/x-aiff Invalid Message Length Content Type audio/x-aiff Verification Failed
12638 0 12638 1 12638 2	Content Type audio/x-ogg Header Check Content Type audio/x-ogg Invalid Message Length Content Type audio/x-ogg Verification Failed
12639 0 12639 1 12639 2	Content Type audio/x-wav Header Check Content Type audio/x-wav Invalid Message Length Content Type audio/x-wav Verification Failed
12641 0 12641 1 12641 2	Content Type text/html Header Check Content Type text/html Invalid Message Length Content Type text/html Verification Failed
12642 0 12642 1	Content Type text/css Header Check Content Type text/css Invalid Message Length
12643 0 12643 1	Content Type text/plain Header Check Content Type text/plain Invalid Message Length
12644 0 12644 1	Content Type text/richtext Header Check Content Type text/richtext Invalid Message Length
12645 0 12645 1 12645 2	Content Type text/sgml Header Check Content Type text/sgml Invalid Message Length Content Type text/sgml Verification Failed
12646 0 12646 1 12646 2	Content Type text/xml Header Check Content Type text/xml Invalid Message Length Content Type text/xml Verification Failed
12648 0 12648 1 12648 2	Content Type video/flc Header Check Content Type video/flc Invalid Message Length Content Type video/flc Verification Failed
12649 0 12649 1 12649 2	Content Type video/mpeg Header Check Content Type video/mpeg Invalid Message Length Content Type video/mpeg Verification Failed

Table 7-2 Define Content Type Signatures (continued)

Signature ID	Signature Description
12650 0 12650 1	Content Type text/xmcd Header Check Content Type text/xmcd Invalid Message Length
12651 0 12651 1 12651 2	Content Type video/quicktime Header Check Content Type video/quicktime Invalid Message Length Content Type video/quicktime Verification Failed
12652 0 12652 1	Content Type video/sgi Header Check Content Type video/sgi Verification Failed
12653 0 12653 1	Content Type video/x-avi Header Check Content Type video/x-avi Invalid Message Length
12654 0 12654 1 12654 2	Content Type video/x-fli Header Check Content Type video/x-fli Invalid Message Length Content Type video/x-fli Verification Failed
12655 0 12655 1 12655 2	Content Type video/x-mng Header Check Content Type video/x-mng Invalid Message Length Content Type video/x-mng Verification Failed
12656 0 12656 1 12656 2	Content Type application/x-msvideo Header Check Content Type application/x-msvideo Invalid Message Length Content Type application/x-msvideo Verification Failed
12658 0 12658 1	Content Type application/ms-word Header Check Content Type application/ms-word Invalid Message Length
12659 0 12659 1	Content Type application/octet-stream Header Check Content Type application/octet-stream Invalid Message Length
12660 0 12660 1 12660 2	Content Type application/postscript Header Check Content Type application/postscript Invalid Message Length Content Type application/postscript Verification Failed
12661 0 12661 1	Content Type application/vnd.ms-excel Header Check Content Type application/vnd.ms-excel Invalid Message Length
12662 0 12662 1	Content Type application/vnd.ms-powerpoint Header Check Content Type application/vnd.ms-powerpoint Invalid Message Length
12663 0 12663 1 12663 2	Content Type application/zip Header Check Content Type application/zip Invalid Message Length Content Type application/zip Verification Failed
12664 0 12664 1 12664 2	Content Type application/x-gzip Header Check Content Type application/x-gzip Invalid Message Length Content Type application/x-gzip Verification Failed
12665 0 12665 1	Content Type application/x-java-archive Header Check Content Type application/x-java-archive Invalid Message Length
12666 0 12666 1	Content Type application/x-java-vm Header Check Content Type application/x-java-vm Invalid Message Length
12667 0 12667 1 12667 2	Content Type application/pdf Header Check Content Type application/pdf Invalid Message Length Content Type application/pdf Verification Failed

Table 7-2 Define Content Type Signatures (continued)

Signature ID	Signature Description
12668 0 12668 1	Content Type unknown Header Check Content Type unknown Invalid Message Length
12669 0 12669 1	Content Type image/x-bitmap Header Check Content Type image/x-bitmap Invalid Message Length
12673 0	Recognized content type

AIC Transfer Encoding Signatures

There are three policies associated with transfer encoding:

- Associate an action with each method (Define Transfer Encoding)
- List methods recognized by the sensor (Recognized Transfer Encodings)
- Specify which actions need to be taken when a chunked encoding error is seen (Chunked Transfer Encoding Error)

Table 7-3 lists the predefined transfer encoding signatures. Enable the signatures that have the predefined transfer encoding method you need. For the procedure for enabling signatures, see Configuring the Status of Signatures, page 7-10.

Table 7-3 Transfer Encoding Signatures

Signature ID	Transfer Encoding Method
12686	Recognized Transfer Encoding
12687	Define Transfer Encoding Deflate
12688	Define Transfer Encoding Identity
12689	Define Transfer Encoding Compress
12690	Define Transfer Encoding GZIP
12693	Define Transfer Encoding Chunked
12694	Chunked Transfer Encoding Error

AIC FTP Commands Signatures

Table 7-4 lists the predefined FTP commands signatures. Enable the signatures that have the predefined FTP command you need. For the procedure for enabling signatures, see Configuring the Status of Signatures, page 7-10.

Table 7-4 FTP Commands Signatures

Signature ID	FTP Command
12900	Unrecognized FTP command
12901	Define FTP command abor
12902	Define FTP command acct
12903	Define FTP command allo
12904	Define FTP command appe

Table 7-4 FTP Commands Signatures (continued)

Signature ID	FTP Command
12905	Define FTP command cdup
12906	Define FTP command cwd
12907	Define FTP command dele
12908	Define FTP command help
12909	Define FTP command list
12910	Define FTP command mkd
12911	Define FTP command mode
12912	Define FTP command nlst
12913	Define FTP command noop
12914	Define FTP command pass
12915	Define FTP command pasv
12916	Define FTP command port
12917	Define FTP command pwd
12918	Define FTP command quit
12919	Define FTP command rein
12920	Define FTP command rest
12921	Define FTP command retr
12922	Define FTP command rmd
12923	Define FTP command rnfr
12924	Define FTP command rnto
12925	Define FTP command site
12926	Define FTP command smnt
12927	Define FTP command stat
12928	Define FTP command stor
12929	Define FTP command stou
12930	Define FTP command stru
12931	Define FTP command syst
12932	Define FTP command type
12933	Define FTP command user

Configuring IP Fragment Reassembly

This section describes IP fragment reassembly, lists the IP fragment reassembly signatures with the configurable parameters, describes how to configure these parameters, and how to configure the method for IP fragment reassembly. It contains the following topics:

- Overview, page 7-22
- IP Fragment Reassembly Signatures and Configurable Parameters, page 7-22

Overview

You can configure the sensor to reassemble a datagram that has been fragmented over multiple packets. You can specify boundaries that the sensor uses to determine how many datagram fragments it reassemble and how long to wait for more fragments of a datagram. The goal is to ensure that the sensor does not allocate all its resources to datagrams that cannot be completely reassembled, either because the sensor missed some frame transmissions or because an attack has been launched that is based on generating random fragmented datagrams.

You configure the IP fragment reassembly per signature.

IP Fragment Reassembly Signatures and Configurable Parameters

Table 7-5 lists IP fragment reassembly signatures with the parameters that you can configure for IP fragment reassembly. The IP fragment reassembly signatures are part of the Normalizer engine.

Table 7-5 IP Fragment Reassembly Signatures

IP Fragment Reassembly Signature	Parameter With Default Value
1200 IP Fragmentation Buffer Full	Specify Max Fragments 10000
1201 IP Fragment Overlap	None
1202 IP Fragment Overrun - Datagram Too Long	Specify Max Datagram Size 65536
1203 IP Fragment Overwrite - Data is Overwritten	None
1204 IP Fragment Missing Initial Fragment	None
1205 IP Fragment Too Many Datagrams	Specify Max Partial Datagrams 1000
1206 IP Fragment Too Small	Specify Max Small Frags 2 Specify Min Fragment Size 400
1207 IP Fragment Too Many Datagrams	Specify Max Fragments per Datagram 170
1208 IP Fragment Incomplete Datagram	Specify Fragment Reassembly Timeout 60
1220 Jolt2 Fragment Reassembly DoS attack	Specify Max Last Fragments 4
1225 Fragment Flags Invalid	None

Configuring IP Fragment Reassembly Parameters

To configure IP fragment reassembly parameters for a specific signature, follow these steps:

Step 1 Log in to the CLI using an account with administrator or operator privileges.

Step 2 Enter signature definition submode:

```
sensor# configure terminal
sensor(config)# service signature-definition sig0
```

Step 3 Specify the IP fragment reassembly signature ID and subsignature ID:

```
sensor(config-sig)# signatures 1200 0
```

Step 4 Specify the engine:

```
sensor(config-sig-sig)# engine normalizer
```

Step 5 Enter edit default signatures submode:

```
sensor(config-sig-sig-nor)# edit-default-sigs-only default-signatures-only
```

Step 6 Enable and change the default setting (if desired) of any of the IP fragment reassembly parameter for signature 1200 for example, specifying the maximum fragments:

```
sensor(config-sig-sig-nor-def)# specify-max-fragments yes
sensor(config-sig-sig-nor-def-yes)# max-fragments 20000
```

Step 7 Verify the settings:

```
sensor(config-sig-sig-nor-def-yes)# show settings
   yes
   ----------------------------------------------
      max-fragments: 20000 default: 10000
   ----------------------------------------------
sensor(config-sig-sig-nor-def-yes)#
```

Step 8 Exit signature definition submode:

```
sensor(config-sig-sig-nor-def-yes)# exit
sensor(config-sig-sig-nor-def)# exit
sensor(config-sig-sig-nor)# exit
sensor(config-sig-sig)# exit
sensor(config-sig)# exit
Apply Changes:?[yes]:
```

Step 9 Press **Enter** for apply the changes or enter **no** to discard them.

Configuring the Method for IP Fragment Reassembly

Use the **fragment-reassembly** command in the signature definition submode to configure the method the sensor will use to reassemble fragments. You can configure this option if your sensor is operating in promiscuous mode. If your sensor is operating in line mode, the method is NT only.

The following options apply:

- **ip-reassemble-mode**—Identifies the method the sensor uses to reassemble the fragments based on the operating system.
  - **nt**—Windows systems.
  - **solaris**—Solaris systems.
  - **linux**—GNU/Linux systems.
  - **bsd**—BSD UNIX systems.

    The default is nt.

To configure IP fragment reassembly, follow these steps:

Step 1 Log in to the CLI using an account with administrator or operator privileges.

Step 2 Enter fragment reassembly submode:

```
sensor# configure terminal
```

```
sensor(config)# service signature-definition sig0
sensor(config-sig)# fragment-reassembly
```

Step 3 Configure the operating system you want the sensor to use to reassemble IP fragments:

```
sensor(config-sig-fra)# ip-reassemble-mode linux
```

Step 4 Verify the setting:

```
sensor(config-sig-fra)# show settings
   fragment-reassembly
   ----------------------------------------------
      ip-reassemble-mode: linux default: nt
   ----------------------------------------------
sensor(config-sig-fra)#
```

Step 5 Exit signature-definition submode:

```
sensor(config-sig-fra)# exit
sensor(config-sig)# exit
Apply Changes:?[yes]:
```

Step 6 Press **Enter** to apply the changes or enter **no** to discard them.

Configuring TCP Stream Reassembly

This section describes TCP stream reassembly, lists the TCP stream reassembly signatures with the configurable parameters, describes how to configure TCP stream signatures, and how to configure the mode for TCP stream reassembly. It contains the following topics:

Overview

You can configure the sensor to monitor only TCP sessions that have been established by a complete three-way handshake. You can also configure how long to wait for the handshake to complete, and how long to keep monitoring a connection where no more packets have been seen. The goal is to prevent the sensor from creating alerts where a valid TCP session has not been established. There are known attacks against sensors that try to get the sensor to generate alerts by simply replaying pieces of an attack. The TCP session reassembly feature helps to mitigate these types of attacks against the sensor.

You configure TCP stream reassembly parameters per signature. You can configure the mode for TCP stream reassembly.

TCP Stream Signatures and Configurable Parameters

Table 7-6 lists TCP stream reassembly signatures with the parameters that you can configure for TCP stream reassembly. TCP stream reassembly signatures are part of the Normalizer engine.

Table 7-6 TCP Stream Reassembly Signatures

Signature ID and Name	Description	Parameter With Default Value and Range	Default Actions
1300 TCP Segment Overwrite[1]	Fires when the data in an overlapping TCP segment (such as a retransmit) sends data that is different from the data already seen on this session	—	Deny Connection Inline Product Alert[2]
1301 TCP Inactive Timeout[3]	Fires when a TCP session has been idle for a TCP Idle Timeout.	TCP Idle Timeout 3600 (15-3600)	None[4]
1302 TCP Embryonic Timeout[5]	Fires when a TCP session has not completes the three-way handshake in TCP embryonic timeout seconds.	TCP Embryonic Timeout 15 (3-300)	None[6]
1303 TCP Closing Timeout[7]	Fires when a TCP session has not closed completely in TCP Closed Timeout seconds after the first FIN.	TCP Closed Timeout 5 (1-60)	None[8]
1304 TCP Max Segments Queued Per Session	Fires when the number of queued out of order segments for a session exceed TCP Max Queue. The segment containing the sequence furthermost from the expected sequence is dropped.	TCP Max Queue 32 (0-128)	Deny Packet Inline Product Alert[9]
1305 TCP Urgent Flag[10]	Fires when the TCP urgent flag is seen	None	Modify Packet Inline is disabled[11]
1306 0 TCP Option Other	Fires when a TCP option in the range of TCP Option Number is seen.	TCP Option Number 6-7,9-255 (Integer Range Allow Multiple 0-255 constraints)	Modify Packet Inline Produce Alert[12]
1306 1 TCP SACK Allowed Option	Fires when a TCP selective ACK allowed option is seen.	—	Modify Packet Inline disabled[13]

Table 7-6 TCP Stream Reassembly Signatures (continued)

Signature ID and Name	Description	Parameter With Default Value and Range	Default Actions
1306 2 TCP SACK Data Option	Fires when a TCP selective ACK data option is seen.	—	Modify Packet Inline disabled[14]
1306 3 TCP Timestamp Option	Fires when a TCP timestamp option is seen.	—	Modify Packet Inline disabled[15]
1306 4 TCP Window Scale Option	Fires when a TCP window scale option is seen.	—	Modify Packet Inline disabled[16]
1307 TCP Window Size Variation	Fires when the right edge of the recv window for TCP moves to the right (decreases).	—	Deny Connection Inline Produce Alert disabled[17]
1308 TTL Varies[18]	Fire when the TTL seen on one direction of a session is higher than the minimum that has been observed	—	Modify Packet Inline[19]
1309 TCP Reserved Bits Set	Fires when the reserved bits (including bits used for ECN) are set on the TCP header.	—	Modify Packet Inline Produce Alert disabled[20]
1310 TCP Retransmit Protection[21]	Fires when the sensor detects that a retransmitted segment has different data than the original segment.	—	Deny Connection Inline Produce Alert[22]
1311 TCP Packet Exceeds MSS	Fires when a packet exceeds the MSS that was exchanged during the three-way handshake.	—	Deny Connection Inline Produce Alert[23]
1312 TCP Min MSS	Fires when the MSS value in a packet containing a SYN flag is less that TCP Min MSS.	TCP Min MSS 400 (0-16000)	Modify Packet Inline disabled[24]
1313 TCP Max MSS	Fires when the MSS value in a packet containing a SYN flag exceed TCP Max MSS	TCP Max MSS1460 (0-16000)	Modify Packet Inline disabled[25]
1314 TCP Data SYN	Fires when TCP payload is sent in the SYN packet.	—	Deny Packet Inline disabled[26]
1315 ACK Without TCP Stream	Fires when an ACK packet is sent that does not belong to a stream.	—	Produce Alert disabled[27]

Table 7-6 TCP Stream Reassembly Signatures (continued)

Signature ID and Name	Description	Parameter With Default Value and Range	Default Actions
1317 Zero Window Probe	Fires when a zero window probe packet is detected.	Modify Packet Inline removes data from the Zero Window Probe packet.	Modify Packet Inline
1330[28] 0 TCP Drop - Bad Checksum	Fires when TCP packet has bad checksum.	Modify Packet Inline corrects the checksum.	Deny Packet Inline
1330 1 TCP Drop - Bad TCP Flags	Fires when TCP packet has bad flag combination.	—	Deny Packet Inline
1330 2 TCP Drop - Urgent Pointer With No Flag	Fires when TCP packet has a URG pointer and no URG flag.	Modify Packet Inline clears the pointer.	Modify Packet Inline disabled
1330 3 TCP Drop - Bad Option List	Fires when TCP packet has a bad option list.	—	Deny Packet Inline
1330 4 TCP Drop - Bad Option Length	Fires when TCP packet has a bad option length.	—	Deny Packet Inline
1330 5 TCP Drop - MSS Option Without SYN	Fires when TCP MSS option is seen in packet without the SYN flag set.	Modify Packet Inline clears the MSS option.	Modify Packet Inline
1330 6 TCP Drop - WinScale Option Without SYN	Fires when TCP window scale option is seen in packet without the SYN flag set.	Modify Packet Inline clears the window scale option.	Modify Packet Inline
1330 7 TCP Drop - Bad WinScale Option Value	Fires when a TCP packet has a bad window scale value.	Modify Packet Inline sets the value to the closest constraint value.	Modify Packet Inline
1330 8 TCP Drop - SACK Allow Without SYN	Fires when the TCP SACK allowed option is seen in a packet without the SYN flags set.	Modify Packet Inline clears the SACK allowed option.	Modify Packet Inline
1330 9 TCP Drop - Data in SYN|ACK	Fires when TCP packet with SYN and ACK flags set also contains data.	—	Deny Packet Inline
1330 10 TCP Drop - Data Past FIN	Fires when TCP data is sequenced after FIN.	—	Deny Packet Inline
1330 11 TCP Drop - Timestamp not Allowed	Fires when TCP packet has timestamp option when timestamp option is not allowed.	—	Deny Packet Inline
1330 12 TCP Drop - Segment Out of Order	Fires when TCP segment is out of order and cannot be queued.	—	Deny Packet Inline

Table 7-6 TCP Stream Reassembly Signatures (continued)

Signature ID and Name	Description	Parameter With Default Value and Range	Default Actions
1330 13 TCP Drop - Invalid TCP Packet	Fires when TCP packet has invalid header.	—	Deny Packet Inline
1330 14 TCP Drop - RST or SYN in window	Fires when TCP packet with RST or SYN flag was sent in the sequence window but was not the next sequence.	—	Deny Packet Inline
1330 15 TCP Drop - Segment Already ACKed	Fires when TCP packet sequence is already ACKed by peer (excluding keepalives).	—	Deny Packet Inline
1330 16 TCP Drop - PAWS Failed	Fires when TCP packet fails PAWS check.	—	Deny Packet Inline
1330 17 TCP Drop - Segment out of State Order	Fires when TCP packet is not proper for the TCP session state.	—	Deny Packet Inline
1330 18 TCP Drop - Segment out of Window	Fires when TCP packet sequence number is outside of allowed window.	—	Deny Packet Inline
3050 Half Open SYN Attack		syn-flood-max-embryonic 5000	
3250 TCP Hijack		max-old-ack 200	
3251 TCP Hijack Simplex Mode		max-old-ack 100	

1. IPS keeps the last 256 bytes in each direction of the TCP session.
2. Modify Packet Inline has no effect on this signature. Deny Connection Inline drops the current packet and the TCP session. Deny Packet Inline drops the packet.
3. The timer is reset to 0 after each packet on the TCP session. by default, this signature does not produce an alert. You can choose to produce alerts for expiring TCP connections if desired. A statistic of total number of expired flows is updated any time a flow expires.
4. Modify Packet Inline, Deny Connection Inline, and Deny Packet Inline have no effect on this signature.
5. The timer starts with the first SYN packet and is not reset. State for the session is reset and any subsequent packets for this flow appear to be out of order (unless it is a SYN).
6. Modify Packet Inline, Deny Connection Inline, and Deny Packet Inline have no effect on this signature.
7. The timer starts with the first FIN packet and is not reset. State for the session is reset and any subsequent packets for this flow appear to be out of order (unless it is a SYN).
8. Modify Packet Inline, Deny Connection Inline, and Deny Packet Inline have no effect on this signature.
9. Modify Packet Inline and Deny Packet Inline have no effect on this signature. Deny Connection Inline drops the current packet and the TCP session.
10. Phrak 57 describes a way to evade security policy using URG pointers. You can normalize the packet when it is in inline mode with this signature.
11. Modify Packet Inline strips the URG flag and zeros the URG pointer from the packet. Deny Connection Inline drops the current packet and the TCP session. Deny Packet Inline drops the packet.
12. Modify Packet Inline strips the selected option(s) from the packet. Deny Connection Inline drops the current packet and the TCP session. Deny Packet Inline drops the packet.
13. Modify Packet Inline strips the selected ACK allowed option from the packet. Deny Connection Inline drops the current packet and the TCP session. Deny Packet Inline drops the packet.

14. Modify Packet Inline strips the selected ACK allowed option from the packet. Deny Connection Inline drops the current packet and the TCP session. Deny Packet Inline drops the packet.
15. Modify Packet Inline strips the timestamp option from the packet. Deny Connection Inline drops the current packet and the TCP session. Deny Packet Inline drops the packet.
16. Modify Packet Inline strips the window scale option from the packet. Deny Connection Inline drops the current packet and the TCP session. Deny Packet Inline drops the packet.
17. Modify Packet Inline has no effect on this signature. Deny Connection Inline drops the current packet and the TCP connection. Deny Packet Inline drops the packet.
18. This signature is used to cause TTLs to monotonically decrease for each direction on a session. For example, if TTL 45 is the lowest TTL seen from A to B, then all future packets from A to B will have a maximum of 45 if Modify Packet Inline is set. Each new low TTL becomes the new maximum for packets on that session.
19. Modify Packet Inline ensures that the IP TTL monotonically decreases. Deny Connection Inline drops the current packet and the TCP session. Deny Packet Inline drops the packet.
20. Modify Packet Inline clears all reserved TCP flags. Deny Connection Inline drops the current packet and the TCP session. Deny Packet Inline drops the packet.
21. This signature is not limited to the last 256 bytes like signature 1300.
22. Modify Packet Inline has no effect on this signature. Deny Connection Inline drops the current packet and the TCP connection. Deny Packet Inline drops the packet.
23. Modify Packet Inline has no effect on this signature. Deny Connection Inline drops the current packet and the TCP connection. Deny Packet Inline drops the packet.
24. 2.4.21-15.EL.cisco.1 Modify Packet Inline raises the MSS value to TCP Min MSS. Deny Connection Inline drops the current packet and the TCP session. Deny Packet Inline drops the packet 2.4.21-15.EL.cisco.1.
25. Modify Packet Inline lowers the MSS value to TCP Max MSS. Deny Connection Inline drops the current packet and the TCP session. Deny Packet Inline drops the packet 2.4.21-15.EL.cisco.1.
26. Modify Packet Inline has no effect on this signature. Deny Connection Inline drops the current packet and the TCP session. Deny Packet Inline drops the packet.
27. Modify Packet Inline, Deny Connection Inline, and Deny Packet Inline have no effect on this signature. By default, the 1330 signatures drop packets for which this signature sends alerts.
28. These subsignatures represent the reasons why the Normalizer might drop a TCP packet. By default these subsignatures drop packets. These subsignatures let you permit packets that fail the checks in the Normalizer through the IPS. The drop reasons have an entry in the TCP statistics. By default these subsignatures do not produce an alert.

Configuring TCP Stream Reassembly Signatures

To configure TCP stream reassembly for a specific signature, follow these steps:

Step 1 Log in to the CLI using an account with administrator or operator privileges.

Step 2 Enter signature definition submode:

```
sensor# configure terminal
sensor(config)# service signature-definition sig0
```

Step 3 Specify the TCP stream reassembly signature ID and subsignature ID:

```
sensor(config-sig)# signatures 1313 0
```

Step 4 Specify the engine:

```
sensor(config-sig-sig)# engine normalizer
```

Step 5 Enter edit default signatures submode:

```
sensor(config-sig-sig-nor)# edit-default-sigs-only default-signatures-only
```

Step 6 Enable and change the default setting (if desired) of the maximum MSS parameter for signature 1313:

```
sensor(config-sig-sig-nor-def)# specify-tcp-max-mss yes
sensor(config-sig-sig-nor-def-yes)# tcp-max-mss 1380
```

Note Changing this parameter from the default of 1460 to 1380 helps prevent fragmentation of traffic going through a VPN tunnel.

Step 7 Verify the settings:

```
sensor(config-sig-sig-nor-def-yes)# show settings
   yes
   -----------------------------------------------
      tcp-max-mss: 1380 default: 1460
   -----------------------------------------------
sensor(config-sig-sig-nor-def-yes)#
```

Step 8 Exit signature definition submode:

```
sensor(config-sig-sig-nor-def-yes)# exit
sensor(config-sig-sig-nor-def)# exit
sensor(config-sig-sig-nor)# exit
sensor(config-sig-sig)# exit
sensor(config-sig)# exit
Apply Changes:?[yes]:
```

Step 9 Press **Enter** for apply the changes or enter **no** to discard them.

Configuring the Mode for TCP Stream Reassembly

Use the **stream-reassembly** command in the signature definition submode to configure the mode that the sensor will use to reassemble TCP sessions.

The following options apply:

- **tcp-3-way-handshake-required {true | false}**—Specifies that the sensor should only track sessions for which the 3-way handshake is completed.

  The default is true.
- **tcp-reassembly-mode**—Specifies the mode the sensor should use to reassemble TCP sessions.
  - **strict**—Only allows the next expected in the sequence.
  - **loose**—Allows gaps in the sequence.
  - **asym**—Allows asymmetric traffic to be reassembled.

    The default is strict.

Caution The asymmetric option disables TCP window evasion checking.

To configure the TCP stream reassembly parameters, follow these steps:

Step 1 Log in to the CLI using an account with administrator or operator privileges.

Step 2 Enter TCP stream reassembly submode:

```
sensor# configure terminal
sensor(config)# service signature-definition sig0
sensor(config-sig)# stream-reassembly
```

Step 3 Specify that the sensor should only track session for which the 3-way handshake is completed:

```
sensor(config-sig-str)# tcp-3-way-handshake-required true
```

Step 4 Specify the mode the sensor should use to reassemble TCP sessions:

```
sensor(config-sig-str)# tcp-reassembly-mode strict
```

Step 5 Verify the settings:

```
sensor(config-sig-str)# show settings
   stream-reassembly
   ----------------------------------------------
      tcp-3-way-handshake-required: true default: true
      tcp-reassembly-mode: strict default: strict
   ----------------------------------------------
sensor(config-sig-str)#
```

Step 6 Exit TCP reassembly submode:

```
sensor(config-sig-str)# exit
sensor(config-sig)# exit
Apply Changes:?[yes]:
```

Step 7 Press **Enter** to apply the changes or enter **no** to discard them.

Configuring IP Logging

You can configure a sensor to generate an IP session log when the sensor detects an attack. When IP logging is configured as a response action for a signature and the signature is triggered, all packets to and from the source address of the alert are logged for a specified period of time.

Use the **ip-log** command in the signature definition submode to configure IP logging.

The following options apply:

- **ip-log-bytes**—Identifies the maximum number of bytes you want logged.

  The valid value is 0 to 2147483647. The default is 0.
- **ip-log-packets**—Identifies the number of packets you want logged.

  The valid value is 0 to 65535. The default is 0.
- **ip-log-time**—Identifies the duration you want the sensor to log.

  The valid value is 30 to 300 seconds. The default is 30 seconds.

Note When the sensor meets any one of the IP logging conditions, it stops IP logging.

To configure the IP logging parameters, follow these steps:

Step 1 Log in to the CLI using an account with administrator or operator privileges.

Step 2 Enter IP log submode:

```
sensor# configure terminal
sensor(config)# service signature-definition sig0
sensor(config-sig)# ip-log
```

Step 3 Specify the IP logging parameters:

a. Specify the maximum number of bytes you want logged:

```
sensor(config-sig-ip)# ip-log-bytes 200000
```

b. Specify the number of packets you want logged:

```
sensor(config-sig-ip)# ip-log-packets 150
```

c. Specify the length of time you want the sensor to log:

```
sensor(config-sig-ip)# ip-log-time 60
```

Step 4 Verify the settings:

```
sensor(config-sig-ip)# show settings
   ip-log
   -----------------------------------------------
      ip-log-packets: 150 default: 0
      ip-log-time: 60 default: 30
      ip-log-bytes: 200000 default: 0
   -----------------------------------------------
sensor(config-sig-ip)#
```

Step 5 Exit IP log submode:

```
sensor(config-sig-ip)# exit
sensor(config-sig)# exit
Apply Changes:?[yes]:
```

Step 6 Press **Enter** to apply the changes or enter **no** to discard them.

Creating Custom Signatures

This section describes how to create custom signatures, and contains the following topics:

Sequence for Creating a Custom Signature

Use the following sequence when you create a custom signature:

Step 1 Select a signature engine.

Step 2 Assign the signature identifiers:

- Signature ID
- SubSignature ID
- Signature name
- Alert notes (optional)
- User comments (optional)

Step 3 Assign the engine-specific parameters.

The parameters differ for each signature engine, although there is a group of master parameters that applies to each engine.

Step 4 Assign the alert response:

- Signature fidelity rating
- Severity of the alert

Step 5 Assign the alert behavior.

Step 6 Apply the changes.

Example String TCP Signature

The String engine is a generic-based pattern-matching inspection engine for ICMP, TCP, and UDP protocols. The String engine uses a regular expression engine that can combine multiple patterns into a single pattern-matching table allowing for a single search through the data.

There are three String engines: String ICMP, String TCP, and String UDP.

The following example demonstrates how to create a custom String TCP signature.

Note This procedure also applies to String UDP and ICMP signatures.

The following parameters apply to the String TCP engine:

- **default**—Sets the value back to the system default setting.
- **direction**—Direction of the traffic:
  - **from-service**—Traffic from service port destined to client port.
  - **to-service**—Traffic from client port destined to service port.
- **event-action**—Action(s) to perform when alert is triggered:
  - **deny-attacker-inline** —(Inline mode only) does not transmit this packet and future packets from the attacker address for a specified period of time.

- **deny-attacker-service-pair-inline**—(Inline only) Does not transmit this packet and future packets on the attacker address victim port pair for a specified period of time.
- **deny-attacker-victim-pair-inline**—(Inline only) Does not transmit this packet and future packets on the attacker/victim address pair for a specified period of time.
- **deny-connection-inline**—(Inline mode only) Does not transmit this packet and future packets on the TCP Flow.
- **deny-packet-inline**—(Inline mode only) Does not transmit this packet.
- **log-attacker-packets**—Starts IP logging of packets containing the attacker address. This action causes an alert to be written to the Event Store, even if **produce-alert** is not selected.
- **log-pair-packets**—Starts IP logging of packets containing the attacker-victim address pair. This action causes an alert to be written to the Event Store, even if **produce-alert** is not selected.
- **log-victim-packets**—Starts IP logging of packets containing the victim address. This action causes an alert to be written to the Event Store, even if **produce-alert** is not selected.
- **produce-alert** —Writes the event to the Event Store as an alert.
- **produce-verbose-alert**—Includes an encoded dump (possibly truncated) of the offending packet in the alert. This action causes an alert to be written to the Event Store, even if **produce-alert** is not selected.
- **request-block-connection**—Sends a request to ARC to block this connection. You must have blocking devices configured to implement this action.
- **request-block-host**—Sends a request to ARC to block this attacker host. You must have blocking devices configured to implement this action.
- **request-rate-limit**—Sends a rate limit request to ARC to perform rate limiting. You must have rate limiting devices configured to implement this action.
- **request-snmp-trap**—Sends a request to the Notification Application component of the sensor to perform SNMP notification. This action causes an alert to be written to the Event Store, even if **produce-alert** is not selected. You must have SNMP configured on the sensor to implement this action.
- **reset-tcp-connection**—Sends TCP resets to hijack and terminate the TCP flow. **Reset TCP Connection** only works on TCP signatures that analyze a single connection. It does not work for sweeps or floods.
- **modify-packet-inline**— Modifies packet data to remove ambiguity about what the end point might do with the packet.

- **no**—Removes an entry or selection setting.
- **regex-string** —A regular expression to search for in a single TCP packet.
- **service-ports**—Ports or port ranges where the target service may reside.

  The valid range is 0 to 65535. It is a separated list of integer ranges a-b[,c-d] within 0 to 65535. The second number in the range must be greater than or equal to the first number.
- **specify-exact-match-offset {yes | no}**—(Optional) Enables exact-match-offset.
- **specify-min-match-length {yes | no}**—(Optional) Enables min-match-length.
- **strip-telnet-options**—Strips Telnet option characters from data before searching.
- **swap-attacker-victim {true | false}**—Swaps the attacker and victim addresses and ports (source and destination) in the alert message and for any actions taken. The default is false for no swapping.

To create a signature based on the String TCP engine, follow these steps:

Step 1 Log in to the CLI using an account with administrator or operator privileges.

Step 2 Enter signature definition submode:

```
sensor# configure terminal
sensor(config)# service signature-definition sig0
```

Step 3 Specify a signature ID and subsignature ID for the signature:

```
sensor(config-sig)# signatures 60025 0
```

Custom signatures are in the range of 60000 to 65000.

Step 4 Enter signature description submode:

```
sensor(config-sig-sig)# sig-description
```

Step 5 Specify a name for the new signature:

```
sensor(config-sig-sig-sig)# sig-name This is my new name
```

You can also specify a additional comments about the sig using the **sig-comment** command or additional information about the signature using the **sig-string-info** command.

Step 6 Exit signature description submode:

```
sensor(config-sig-sig-sig)# exit
```

Step 7 Specify the string TCP engine:

```
sensor(config-sig-sig)# engine string-tcp
```

Step 8 Specify the service ports:

```
sensor(config-sig-sig-str)# service-ports 23
```

Step 9 Specify the direction:

```
sensor(config-sig-sig-str)# direction to-service
```

Step 10 Specify the regex string to search for in the TCP packet:

```
sensor(config-sig-sig-str)# regex-string This-is-my-new-Sig-regex
```

Step 11 You can change the event actions if needed according to your security policy using the **event-action** command. The default event action is **produce-alert**.

Step 12 You can modify the following optional parameters for this custom String TCP signature:

- **specify-exact-match-offset**
- **specify-min-match-length**
- **strip-telnet-options**
- **swap-attacker-victim**.

Step 13 Verify the settings:

```
sensor(config-sig-sig-str)# show settings
   string-tcp
   -----------------------------------------------
      event-action: produce-alert <defaulted>
      strip-telnet-options: false <defaulted>
      specify-min-match-length
      -----------------------------------------------
```

```
            no
            ----------------------------------------------
            ----------------------------------------------
         ---------------------------------------------
         regex-string: This-is-my-new-Sig-regex
         service-ports: 23
         direction: to-service default: to-service
         specify-exact-match-offset
         ---------------------------------------------
            no
            ----------------------------------------------
               specify-max-match-offset
               ----------------------------------------------
                  no
                  ----------------------------------------------
                  ----------------------------------------------
               ----------------------------------------------
               specify-min-match-offset
               ----------------------------------------------
                  no
                  ----------------------------------------------
                  ----------------------------------------------
               ----------------------------------------------
            ----------------------------------------------
         ---------------------------------------------
         swap-attacker-victim: false <defaulted>
      ---------------------------------------------
sensor(config-sig-sig-str)#
```

Step 14 Exit signature definition submode:

```
sensor(config-sig-sig-str)# exit
sensor(config-sig-sig)# exit
sensor(config-sig)# exit
Apply Changes:?[yes]:
```

Step 15 Press **Enter** to apply the changes or enter **no** to discard them.

Example Service HTTP Signature

The Service HTTP engine is a service-specific string-based pattern-matching inspection engine. The HTTP protocol is one of the most commonly used in today's networks. In addition, it requires the most amount of preprocessing time and has the most number of signatures requiring inspection making it critical to the system's overall performance.

The Service HTTP engine uses a Regex library that can combine multiple patterns into a single pattern-matching table allowing a single search through the data. This engine searches traffic directed to web services only to web services, or HTTP requests. You cannot inspect return traffic with this engine. You can specify separate web ports of interest in each signature in this engine.

HTTP deobfuscation is the process of decoding an HTTP message by normalizing encoded characters to ASCII equivalent characters. It is also known as ASCII normalization.

Before an HTTP packet can be inspected, the data must be deobfuscated or normalized to the same representation that the target system sees when it processes the data. It is ideal to have a customized decoding technique for each host target type, which involves knowing what operating system and web server version is running on the target. The Service HTTP engine has default deobfuscation behavior for the Microsoft IIS web server.

The following example demonstrates how to create a custom Service HTTP signature.

The following options apply to the Service HTTP engine:

- **de-obfuscate {true | false}**—Applies anti-evasive deobfuscation before searching.
- **default**—Sets the value back to the system default setting.
- **event-action** —Action(s) to perform when alert is triggered:
  - **deny-attacker-inline** —(Inline mode only) does not transmit this packet and future packets from the attacker address for a specified period of time.
  - **deny-attacker-service-pair-inline**—(Inline only) Does not transmit this packet and future packets on the attacker address victim port pair for a specified period of time.
  - **deny-attacker-victim-pair-inline**—(Inline only) Does not transmit this packet and future packets on the attacker/victim address pair for a specified period of time.
  - **deny-connection-inline**—(Inline mode only) Does not transmit this packet and future packets on the TCP Flow.
  - **deny-packet-inline**—(Inline mode only) Does not transmit this packet.
  - **log-attacker-packets**—Starts IP logging of packets containing the attacker address. This action causes an alert to be written to the Event Store, even if **produce-alert** is not selected.
  - **log-pair-packets**—Starts IP logging of packets containing the attacker-victim address pair. This action causes an alert to be written to the Event Store, even if **produce-alert** is not selected.
  - **log-victim-packets**—Starts IP logging of packets containing the victim address. This action causes an alert to be written to the Event Store, even if **produce-alert** is not selected.
  - **produce-alert** —Writes the event to the Event Store as an alert.
  - **produce-verbose-alert**—Includes an encoded dump (possibly truncated) of the offending packet in the alert. This action causes an alert to be written to the Event Store, even if **produce-alert** is not selected.
  - **request-block-connection**—Sends a request to ARC to block this connection. You must have blocking devices configured to implement this action.
  - **request-block-host**—Sends a request to ARC to block this attacker host. You must have blocking devices configured to implement this action.
  - **request-rate-limit**—Sends a rate limit request to ARC to perform rate limiting. You must have rate limiting devices configured to implement this action.
  - **request-snmp-trap**—Sends a request to the Notification Application component of the sensor to perform SNMP notification. This action causes an alert to be written to the Event Store, even if **produce-alert** is not selected. You must have SNMP configured on the sensor to implement this action.
  - **reset-tcp-connection**—Sends TCP resets to hijack and terminate the TCP flow. **Reset TCP Connection** only works on TCP signatures that analyze a single connection. It does not work for sweeps or floods.
  - **modify-packet-inline**— Modifies packet data to remove ambiguity about what the end point might do with the packet.
- **max-field-sizes** —Grouping for maximum field sizes:
  - **specify-max-arg-field-length {yes | no}**—Enables max-arg-field-length (optional).
  - **specify-max-header-field-length {yes | no}]**—Enables max-header-field-length (optional).
  - **specify-max-request-length {yes | no}**—Enables max-request-length (optional).

  - **specify-max-uri-field-length {yes | no}**—Enables max-uri-field-length (optional).
- **no**—Removes an entry or selection setting.
- **regex**—Regular expression grouping:
  - **specify-arg-name-regex**—Enables arg-name-regex (optional).
  - **specify-header-regex** —Enables header-regex (optional).
  - **specify-request-regex**—Enables request-regex (optional).
  - **specify-uri-regex**—Enables uri-regex (optional).
- **service-ports** —A comma-separated list of ports or port ranges where the target service may reside.
- **swap-attacker-victim {true | false}**—Swaps the attacker and victim addresses and ports (source and destination) in the alert message and for any actions taken. The default is false for no swapping.

To create a custom signature based on the Service HTTP engine, follow these steps:

Step 1 Log in to the CLI using an account with administrator or operator privileges.

Step 2 Enter signature definition submode:

```
sensor# configure terminal
sensor(config)# service signature-definition sig0
```

Step 3 Specify a signature ID and a subsignature ID for the signature:

```
sensor(config-sig)# signatures 63000 0
```

Custom signatures are in the range of 60000 to 65000.

Step 4 Enter signature description mode:

```
sensor(config-sig-sig)# sig-description
```

Step 5 Specify a signature name:

```
sensor(config-sig-sig-sig)# sig-name myWebSig
```

Step 6 Specify the alert traits:

```
sensor(config-sig-sig-sig)# alert-traits 2
```

The valid range is from 0 to 65535.

Step 7 Exit signature description submode:

```
sensor(config-sig-sig-sig)# exit
```

Step 8 Assign the alert frequency:

```
sensor(config-sig-sig)# alert-frequency
sensor(config-sig-sig-ale)# summary-mode fire-all
sensor(config-sig-sig-ale-fir)# summary-key Axxx
sensor(config-sig-sig-ale-fir)# specify-summary-threshold yes
sensor(config-sig-sig-ale-fir-yes)# summary-threshold 200
```

Step 9 Exit alert frequency submode:

```
sensor(config-sig-sig-ale-fir-yes)# exit
sensor(config-sig-sig-ale-fir)# exit
sensor(config-sig-sig-ale)# exit
```

Step 10 Configure the signature to apply anti-evasive deobfuscation before searching:

```
sensor(config-sig-sig)# engine service-http
sensor(config-sig-sig-ser)# de-obfuscate true
```

Step 11 Configure the regex parameters:

```
sensor(config-sig-sig)# engine service-http
sensor(config-sig-sig-ser)# regex
sensor(config-sig-sig-ser-reg)# specify-uri-regex yes
sensor(config-sig-sig-ser-reg-yes)# uri-regex [Mm][Yy][Ff][Oo][Oo]
```

Step 12 Exit regex submode:

```
sensor(config-sig-sig-ser-reg-yes)# exit
sensor(config-sig-sig-ser-reg-)# exit
```

Step 13 Configure the service ports using the signature variable WEBPORTS:

```
sensor(config-sig-sig-ser)# service-ports $WEBPORTS
```

Step 14 Exit signature definition submode:

```
sensor(config-sig-sig-ser)# exit
sensor(config-sig-sig)# exit
sensor(config-sig)# exit
Apply Changes:?[yes]:
```

Step 15 Press **Enter** to apply the changes or enter **no** to discard them.

Example MEG Signature

The Meta engine defines events that occur in a related manner within a sliding time interval. This engine processes events rather than packets. As signature events are generated, the Meta engine inspects them to determine if they match any or several Meta definitions. The Meta engine generates a signature event after all requirements for the event are met.

All signature events are handed off to the Meta engine by SEAP. SEAP hands off the event after processing the minimum hits option. Summarization and event action are processed after the Meta engine has processed the component events. For more information about SEAP, see Signature Event Action Processor, page 6-2.

Caution A large number of Meta signatures could adversely affect overall sensor performance.

The following example demonstrates how to create a MEG signature based on the Meta engine.

Note The Meta engine is different from other engines in that it takes alerts as input where most engines take packets as input. For more information on the Meta engine, see Meta Engine, page B-13.

The following options apply to the Meta signature engine:

- **component-list**—List of Meta components.
  - **edit**—Edits an existing entry in the list.
  - **insert** *name1*—Inserts a new entry into the list.

  - **move**—Moves an entry in the list.
  - **begin**—Places the entry at the beginning of the active list.
  - **end**—Places the entry at the end of the active list.
  - **inactive**—Places the entry into the inactive list.
  - **before**—Places the entry before the specified entry.
  - **after**—Places the entry after the specified entry.
- **component-count**—Number of times component must fire before this component is satisfied.
- **component-sig-id**—Signature ID of the signature to match this component on.
- **component-subsig-id**—Subsignature ID of the signature to match this component on.
- **component-list-in-order {true | false}**—Whether or not to have the component list fire in order.
- **event-action**—Action(s) to perform when alert is triggered:
  - **deny-attacker-inline** —(Inline mode only) does not transmit this packet and future packets from the attacker address for a specified period of time.
  - **deny-attacker-service-pair-inline**—(Inline only) Does not transmit this packet and future packets on the attacker address victim port pair for a specified period of time.
  - **deny-attacker-victim-pair-inline**—(Inline only) Does not transmit this packet and future packets on the attacker/victim address pair for a specified period of time.
  - **deny-connection-inline**—(Inline mode only) Does not transmit this packet and future packets on the TCP Flow.
  - **deny-packet-inline**—(Inline mode only) Does not transmit this packet.
  - **log-attacker-packets**—Starts IP logging of packets containing the attacker address. This action causes an alert to be written to the Event Store, even if **produce-alert** is not selected.
  - **log-pair-packets**—Starts IP logging of packets containing the attacker-victim address pair. This action causes an alert to be written to the Event Store, even if **produce-alert** is not selected.
  - **log-victim-packets**—Starts IP logging of packets containing the victim address. This action causes an alert to be written to the Event Store, even if **produce-alert** is not selected.
  - **produce-alert** —Writes the event to the Event Store as an alert.
  - **produce-verbose-alert**—Includes an encoded dump (possibly truncated) of the offending packet in the alert. This action causes an alert to be written to the Event Store, even if **produce-alert** is not selected.
  - **request-block-connection**—Sends a request to ARC to block this connection. You must have blocking devices configured to implement this action.
  - **request-block-host**—Sends a request to ARC to block this attacker host. You must have blocking devices configured to implement this action.
  - **request-rate-limit**—Sends a rate limit request to ARC to perform rate limiting. You must have rate limiting devices configured to implement this action.
  - **request-snmp-trap**—Sends a request to the Notification Application component of the sensor to perform SNMP notification. This action causes an alert to be written to the Event Store, even if **produce-alert** is not selected. You must have SNMP configured on the sensor to implement this action.
  - **reset-tcp-connection**—Sends TCP resets to hijack and terminate the TCP flow. **Reset TCP Connection** only works on TCP signatures that analyze a single connection. It does not work for sweeps or floods.

  - **modify-packet-inline**— Modifies packet data to remove ambiguity about what the end point might do with the packet.
- **meta-key**—Storage type for the Meta signature.
  - **AaBb**—Attacker and victim addresses and ports.
  - **AxBx**—Attacker and victim addresses.
  - **Axxx**—Attacker address.
  - **xxBx**—Victim address.
- **meta-reset-interval**—Time in seconds to reset the Meta signature.

  The valid range is 0 to 3600 seconds. The default is 60 seconds.

Note Signature 64000 subsignature 0 will fire when it sees the alerts from signature 2000 subsignature 0 and signature 3000 subsignature 0 on the same source address. The source address selection is a result of the meta key default value of Axxx. You can change the behavior by changing the meta key setting to xxBx (destination address) for example.

To create a MEG signature based on the Meta engine, follow these steps:

Step 1 Log in to the CLI using an account with administrator or operator privileges.

Step 2 Enter signature definition submode:

```
sensor# configure terminal
sensor(config)# service signature-definition sig0
```

Step 3 Specify a signature ID and a subsignature ID for the signature:

```
sensor(config-sig)# signatures 64000 0
```

Custom signatures are in the range of 60000 to 65000.

Step 4 Specify the signature engine:

```
sensor(config-sig-sig)# engine meta
```

Step 5 Insert a MEG signature (named c1) at the beginning of the list:

```
sensor(config-sig-sig-met)# component-list insert c1 begin
```

Step 6 Specify the signature ID of the signature on which to match this component:

```
sensor(config-sig-sig-met-com)# component-sig-id 2000
```

Step 7 Exit component list submode:

```
sensor(config-sig-sig-met-com)# exit
```

Step 8 Insert another MEG signature (named c2) at the end of the list:

```
sensor(config-sig-sig-met)# component-list insert c2 end
```

Step 9 Specify the signature ID of the signature on which to match this component

```
sensor(config-sig-sig-met-com)# component-sig-id 3000
```

Step 10 Verify the settings:

```
sensor(config-sig-sig-met-com)# exit
sensor(config-sig-sig-met)# show settings
```

```
   meta
   ----------------------------------------------
      event-action: produce-alert <defaulted>
      meta-reset-interval: 60 <defaulted>
      component-list (min: 1, max: 8, current: 2 - 2 active, 0 inactive)
      ----------------------------------------------
      ACTIVE list-contents
      ----------------------------------------------
         NAME: c1
         ----------------------------------------------
            component-sig-id: 2000
            component-subsig-id: 0 <defaulted>
            component-count: 1 <defaulted>
         ----------------------------------------------
         ----------------------------------------------
         NAME: c2
         ----------------------------------------------
            component-sig-id: 3000
            component-subsig-id: 0 <defaulted>
            component-count: 1 <defaulted>
         ----------------------------------------------
         ----------------------------------------------
      ----------------------------------------------
      meta-key
      ----------------------------------------------
         Axxx
         ----------------------------------------------
            unique-victims: 1 <defaulted>
         ----------------------------------------------
      ----------------------------------------------
      component-list-in-order: false <defaulted>
   ----------------------------------------------
sensor(config-sig-sig-met)#
```

Step 11 Exit signature definition submode:

```
sensor(config-sig-sig-met)# exit
sensor(config-sig-sig)# exit
sensor(config-sig)# exit
Apply Changes:?[yes]:
```

Step 12 Press **Enter** to apply the changes or enter **no** to discard them.

Example AIC MIME-Type Signature

The following example demonstrates how to create a MIME-type signature based on the AIC engine.

The following options apply:

- **event-action**—Action(s) to perform when alert is triggered.
  - **deny-attacker-inline** —(Inline mode only) does not transmit this packet and future packets from the attacker address for a specified period of time.
  - **deny-attacker-service-pair-inline**—(Inline only) Does not transmit this packet and future packets on the attacker address victim port pair for a specified period of time.
  - **deny-attacker-victim-pair-inline**—(Inline only) Does not transmit this packet and future packets on the attacker/victim address pair for a specified period of time.

  - **deny-connection-inline**—(Inline mode only) Does not transmit this packet and future packets on the TCP Flow.
  - **deny-packet-inline**—(Inline mode only) Does not transmit this packet.
  - **log-attacker-packets**—Starts IP logging of packets containing the attacker address. This action causes an alert to be written to the Event Store, even if **produce-alert** is not selected.
  - **log-pair-packets**—Starts IP logging of packets containing the attacker-victim address pair. This action causes an alert to be written to the Event Store, even if **produce-alert** is not selected.
  - **log-victim-packets**—Starts IP logging of packets containing the victim address. This action causes an alert to be written to the Event Store, even if **produce-alert** is not selected.
  - **produce-alert** —Writes the event to the Event Store as an alert.
  - **produce-verbose-alert**—Includes an encoded dump (possibly truncated) of the offending packet in the alert. This action causes an alert to be written to the Event Store, even if **produce-alert** is not selected.
  - **request-block-connection**—Sends a request to ARC to block this connection. You must have blocking devices configured to implement this action.
  - **request-block-host**—Sends a request to ARC to block this attacker host. You must have blocking devices configured to implement this action.
  - **request-rate-limit**—Sends a rate limit request to ARC to perform rate limiting. You must have rate limiting devices configured to implement this action.
  - **request-snmp-trap**—Sends a request to the Notification Application component of the sensor to perform SNMP notification. This action causes an alert to be written to the Event Store, even if **produce-alert** is not selected. You must have SNMP configured on the sensor to implement this action.
  - **reset-tcp-connection**—Sends TCP resets to hijack and terminate the TCP flow. **Reset TCP Connection** only works on TCP signatures that analyze a single connection. It does not work for sweeps or floods.
  - **modify-packet-inline**— Modifies packet data to remove ambiguity about what the end point might do with the packet.
- **no**—Removes an entry or selection setting
- **signature-type**—Type of signature desired
  - **content-types**—Content-types
  - **define-web-traffic-policy**—Defines web traffic policy
  - **max-outstanding-requests-overrun**—Inspects for large number of outstanding HTTP requests
  - **msg-body-pattern**—Message body pattern
  - **request-methods**—Signature types that deal with request methods
  - **transfer-encodings**—Signature types that deal with transfer encodings

To define a MIME-type policy signature, follow these steps:

Step 1 Log in to the CLI using an account with administrator or operator privileges.

Step 2 Enter application policy enforcement submode:

```
sensor# configure terminal
sensor(config)# service signature-definition sig0
```

```
sensor(config-sig)# signatures 60001 0
sensor(config-sig-sig)# engine application-policy-enforcement-http
```

Step 3 Specify the event action:

```
sensor(config-sig-sig-app)# event-action produce-alert|log-pair-packets
```

Step 4 Define the signature type:

```
sensor(config-sig-sig-app)# signature-type content-type define-content-type
```

Step 5 Define the content type:

```
sensor(config-sig-sig-app-def)# name MyContent
```

Step 6 Verify your settings:

```
sensor(config-sig-sig-app-def)# show settings
-> define-content-type
   ----------------------------------------------
      name: MyContent
*---> content-type-details
      ----------------------------------------------
      ----------------------------------------------
   ----------------------------------------------
sensor(config-sig-sig-app-def)#
```

Step 7 Exit signatures submode:

```
sensor(config-sig-sig-app-def)# exit
sensor(config-sig-sig-app)# exit
sensor(config-sig-sig)# exit
sensor(config-sig)# exit
Apply Changes:?[yes]:
```

Step 8 Press **Enter** to apply the changes or enter **no** to discard them.

CHAPTER 8

Configuring IP Logging

This chapter describes how to configure IP logging on the sensor. It contains the following sections:

Understanding IP Logging

You can manually configure the sensor to capture all IP traffic associated with a host you specify by IP address. You can specify how long you want the IP traffic to be logged, how many packets you want logged, and how many bytes you want logged. The sensor stops logging IP traffic at the first parameter you specify.

You can also have the sensor log IP packets every time a particular signature is fired. You can specify how long you want the sensor to log IP traffic and how many packets and bytes you want logged.

Caution Turning on IP logging slows down system performance.

Note You cannot delete or manage IP log files. The **no iplog** command does not delete IP logs, it only stops more packets from being recorded for that IP log. IP logs are stored in a circular buffer that is never filled because new IP logs overwrite old ones.

You can copy the IP logs from the sensor and have them analyzed by a tool that can read packet files in a libpcap format, such as Wireshark or TCPDUMP.

Note Each alert references IP logs that are created because of that alert. If multiple alerts create IP logs for the same IP address, only one IP log is created for all the alerts. Each alert references the same IP log. However, the output of the IP log status only shows the event ID of the first alert triggering the IP log.

Configuring Automatic IP Logging

Use the **ip-log-packets** *number*, **ip-log-time** *number*, and **ip-log-bytes** *number* commands to configure automatic IP logging parameters on the sensor.

The following options apply:

- **ip-log-packets**—Identifies the number of packets you want logged.

  The valid value is 0 to 65535. The default is 0.
- **ip-log-time**—Identifies the duration you want the sensor to log packets.

  The valid value is 0 to 65535 minutes. The default is 30 minutes.
- **ip-log-bytes** —Identifies the maximum number of bytes you want logged.

  The valid value is 0 to 2147483647. The default is 0.

Note An automatic IP log continues capturing packets until one of these parameters is reached.

To reset the parameters, use the **default** keyword. To copy and view an IP log file, see Copying IP Log Files to Be Viewed, page 8-5.

Automatic IP logging is configured on a per signature basis or as an event action override. The following actions trigger automatic IP logging:

- log-attacker-packets
- log-victim-packets
- log-pair-packets

For more information, see Chapter 6, "Configuring Event Action Rules."

To configure automatic IP logging parameters, follow these steps:

Step 1 Log in to the CLI using an account with administrator or operator privileges.

Step 2 Enter signature IP log configuration submode:

```
sensor# configure terminal
sensor(config)# service signature-definition sig0
sensor(config-sig)# ip-log
```

Step 3 Configure the number of packets you want the sensor to log:

```
sensor(config-sig-ip)# ip-log-packets 200
```

Step 4 Configure the duration you want the sensor to log packets:

```
sensor(config-sig-ip)# ip-log-time 60
```

Step 5 Configure the number of bytes you want logged:

```
sensor(config-sig-ip)# ip-log-bytes 5024
```

Step 6 Verify the settings:

```
sensor(config-sig-ip)# show settings
   ip-log
   -----------------------------------------------
      ip-log-packets: 200 default: 0
      ip-log-time: 60 default: 30
      ip-log-bytes: 5024 default: 0
```

```
    ----------------------------------------------
sensor(config-sig-ip)#
```

Step 7 Exit IP logging mode:

```
sensor(config-sig-ip)# exit
sensor(config-sig)# exit
Apply Changes?:[yes]:
```

Step 8 Press **Enter** to apply the changes or enter **no** to discard the changes.

Configuring Manual IP Logging for a Specific IP Address

Use the **iplog** *name ip_address* [**duration** *minutes*] [**packets** *numPackets*] [**bytes** *numBytes*] command to log IP packets manually on the virtual sensor for a specific IP address.

The following options apply:

- *name*—Virtual sensor on which to begin and end logging.

  Note There is only one virtual sensor name in IPS 5.0, vs0.

- *ip_address*—Logs packets containing the specified source and/or destination IP address.
- *minutes*—Duration the logging should be active.

  The valid range is 1 to 60 minutes. The default is 10 minutes.
- *numPackets*—Maximum number of packets to log.

  The valid range is 0 to 4294967295. The default is 1000 packets.
- *numBytes*—Maximum number of bytes to log.

  The valid range is 0 to 4294967295. A value of 0 indicates unlimited bytes.

Note The *minutes*, *numPackets*, and *numBytes* parameters are optional, you do not have to specify all three. However, if you include more than one parameter, the sensor continues logging only until the first threshold is reached. For example, if you set the duration to 5 minutes and the number of packets to 1000, the sensor stops logging after the 1000th packet is captured, even if only 2 minutes have passed.

To stop logging IP packets for a specific IP address, see Stopping Active IP Logs, page 8-4. To log IP packets as an event associated with a signature, see Configuring Automatic IP Logging, page 8-2. To copy and view an IP log file, see Copying IP Log Files to Be Viewed, page 8-5.

To manually log packets on the virtual sensor for a specific IP address, follow these steps:

Step 1 Log in to the CLI using an account with administrator or operator privileges.

Step 2 Start IP logging for a specific IP address:

```
sensor# iplog vs0 10.16.0.0 duration 5
Logging started for virtual sensor vs0, IP address 10.16.0.0, Log ID 1
Warning: IP Logging will affect system performance.
sensor#
```

The example shows the sensor logging all IP packets for 5 minutes to and from the IP address 10.16.0.0.

Note Make note of the Log ID for future reference.

Step 3 Monitor the IP log status with the **iplog-status** command:

```
sensor# iplog-status
Log ID:             1
IP Address 1:       10.16.0.0
Virtual Sensor:     vs0
Status:             added
Event ID:           0
Bytes Captured:     0
Packets Captured:   0
sensor#
```

Note Each alert references IP logs that are created because of that alert. If multiple alerts create IP logs for the same IP address, only one IP log is created for all the alerts. Each alert references the same IP log. However, the output of the IP log status only shows the event ID of the first alert triggering the IP log.

Stopping Active IP Logs

Use the **no iplog** [**log-id** *log_id* | **name** *name*] command to stop logging for the logs that are in the `started` state and to remove logs that are in the `added` state.

Note Using the **no iplog** command on an added state IP log stops the IP log. The added state means that the IP log is still empty (no packets). Stopping it when there are no packets means you are stopping an empty IP log. An empty logged is removed when it is stopped.

Note The **no iplog** command does not remove or delete the IP log. It only signals to the sensor to stop capturing additional packets on that IP log.

The following options apply:

- *log_id*—Log ID of the logging session to stop. Use the **iplog-status** command to find the log ID.
- *name*—Virtual sensor on which to begin or end logging.

  Note There is only one virtual sensor name in IPS 5.1, vs0.

To disable one or all IP logging sessions, follow these steps:

Step 1 Log in to the CLI using an account with administrator or operator privileges.

Step 2 To stop a particular IP logging session:

a. Find the log ID of the session you want to stop by using the **iplog-status** command:

```
sensor# iplog-status
Log ID:             1
IP Address 1:       10.16.0.0
Virtual Sensor:     vs0
Status:             added
Event ID:           0
Bytes Captured:     0
Packets Captured:   0
sensor#
```

Note Each alert references IP logs that are created because of that alert. If multiple alerts create IP logs for the same IP address, only one IP log is created for all the alerts. Each alert references the same IP log. However, the output of the IP log status only shows the event ID of the first alert triggering the IP log.

b. Stop the IP log session:

```
sensor# no iplog log-id 137857512
```

Step 3 To stop all IP logging sessions on the virtual sensor:

```
sensor# no iplog name vs0
```

Step 4 Verify that IP logging has been stopped:

```
sensor# iplog-status
Log ID:             1
IP Address 1:       10.16.0.0
Virtual Sensor:     vs0
Status:             completed
Event ID:           0
Bytes Captured:     0
Packets Captured:   0
sensor#
```

When the logs are stopped, the status shows them as completed.

Copying IP Log Files to Be Viewed

Use the **copy iplog** *log_id destination_url* command to copy IP log files to an FTP or SCP server so that you can view them with a sniffing tool such as Wireshark or TCPDUMP.

The following options apply:

- *log_id*—The log ID of the logging session. You can retrieve the log ID using the **iplog-status** command.
- *destination_url*—The location of the destination file to be copied. It can be a URL or a keyword.

The exact format of the source and destination URLs varies according to the file. Here are the valid types:

- ftp:—Destination URL for an FTP network server. The syntax for this prefix is:

  ftp:[//[username@] location]/relativeDirectory]/filename

  ftp:[//[username@]location]//absoluteDirectory]/filename

- scp:—Destination URL for the SCP network server. The syntax for this prefix is:

  scp:[//[username@] location]/relativeDirectory]/filename

  scp:[//[username@] location]//absoluteDirectory]/filename

When you use FTP or SCP protocol, you are prompted for a password.

To copy IP log files to an FTP or SCP server, follow these steps:

Step 1 Log in to the CLI.

Step 2 Monitor the IP log status with the **iplog-status** command until you see that the status reads completed for the log ID of the log file that you want to copy:

```
sensor# iplog-status
Log ID:              2425
IP Address:          10.1.1.2
Virtual Sensor:      vs0
Status:              started
Start Time:          2003/07/30 18:24:18 2002/07/30 12:24:18 CST
Packets Captured:    1039438

Log ID:              2342
IP Address:          10.2.3.1
Virtual Sensor:      vs0
Status:              completed
Event ID:            209348
Start Time:          2003/07/30 18:24:18 2002/07/30 12:24:18 CST
End Time:            2003/07/30 18:34:18 2002/07/30 12:34:18 CST
sensor#
```

Step 3 Copy the IP log to your FTP or SCP server:

```
sensor# copy iplog 2342 ftp://root@10.16.0.0/user/iplog1
Password: ******** Connected to 10.16.0.0 (10.16.0.0). 220 linux.machine.com FTP server
(Version wu-2.6.0(1) Mon Feb 28 10:30:36 EST 2000) ready. ftp> user (username) root 331
Password required for root. Password:230 User root logged in. ftp> 200 Type set to I. ftp>
put iplog.8518.tmp iplog1 local: iplog.8518.tmp remote: iplog1 227 Entering Passive Mode
(2,4,6,8,179,125) 150 Opening BINARY mode data connection for iplog1. 226 Transfer
complete. 30650 bytes sent in 0.00246 secs (1.2e+04 Kbytes/sec) ftp>
```

Step 4 Open the IP log using a sniffer program such as Wireshark or TCPDUMP.

For more information on Wireshark go to http://www.wireshark.org. For more information on TCPDUMP, go to http://www.tcpdump.org/.

CHAPTER 9

Displaying and Capturing Live Traffic on an Interface

This chapter describes how to display, capture, copy, and erase packet files. It contains the following sections:

Understanding Packet Display and Capture

You can display or capture live traffic from an interface and have the live traffic or a previously captured file put directly on the screen. Storage is available for one local file only, subsequent capture requests overwrites an existing file. The size of the storage file varies depending on the platform. A message may be displayed if the maximum file size is reached before the requested packet count is captured.

Note Capturing live traffic off the interface does not disrupt any of the functionality of the sensor.

Caution Changing the interface configuration results in abnormal termination of any **packet** command running on that interface.

Caution Executing the **packet display** or **capture** command causes significant performance degradation.

Displaying Live Traffic on an Interface

Use the **packet display** *interface_name* [**snaplen** *length*] [**count** *count*] [**verbose**] [**expression** *expression*] command to display live traffic from an interface directly on your screen. Use the **packet display iplog** *id* [**verbose**] [**expression** *expression*] to display iplogs.

Note Press **Ctrl-C** to terminate the live display.

The following options apply:

- *interface_name*—Logical interface name.

  You can only use an interface name that exists in the system.
- **snaplen**—Maximum number of bytes captured for each packet (optional).

  The valid range is 68 to 1600. The default is 0. A value of 0 means use the required length to catch whole packets.
- **count**—Maximum number of packets to capture (optional).

  The valid range is 1 to 10000.

  Note If you do not specify this option, the capture terminates after the maximum file size is captured.
- **verbose**—Displays the protocol tree for each packet rather than a one-line summary (optional).
- **expression**—Packet-display filter expression. This expression is passed directly to TCPDUMP and must meet the TCPDUMP expression syntax.

  Note The expression syntax is described in the TCPDUMP man page.

  Note When using the **expression** option when monitoring packets with VLAN headers, the expression does not match properly unless **vlan and** is added to the beginning of the expression. For example, **packet display iplog 926299444 verbose expression icmp** Will NOT show ICMP packets; **packet display iplog 926299444 verbose expression vlan and icmp** WILL show ICMP packets. It is often necessary to use **expression vlan and** on the IDSM2 and IPS appliance interfaces connected to trunk ports.
- **file-info**—Displays information about the stored packet file.

  File-info displays the following information:

  Captured by: *user:id*, Cmd: *cliCmd*

  Start: yyyy/mm/dd hh:mm:ss zone, End: yyyy/mm/dd hh:mm:ss zone or in-progress

  Where

  user = username of user initiating capture

  id = user's CLI ID

  cliCmd = command entered to perform the capture

Caution Executing the **packet display** command causes significant performance degradation.

To configure the sensor to display live traffic from an interface on the screen, follow these steps:

Step 1 Log in to the sensor using an account with administrator or operator privileges.

Step 2 Display the live traffic on the interface you are interested in, for example, GigabitEthernet0/1:

```
sensor# packet display GigabitEthernet0/1
Warning: This command will cause significant performance degradation
tcpdump: listening on ge0_1, link-type EN10MB (Ethernet), capture size 65535 bytes
03:43:05.691883 IP (tos 0x10, ttl  64, id 55460, offset 0, flags [DF], length: 100)
10.89.147.31.22 > 10.89.147.50.41805: P [tcp sum ok] 4233955485:4233955533(48) ack
1495691730 win 8576 <nop,nop,timestamp 44085169 226014949>
03:43:05.691975 IP (tos 0x10, ttl  64, id 55461, offset 0, flags [DF], length: 164)
10.89.147.31.22 > 10.89.147.50.41805: P [tcp sum ok] 48:160(112) ack 1 win 8576
<nop,nop,timestamp 44085169 226014949>
03:43:05.691998 IP (tos 0x10, ttl  64, id 53735, offset 0, flags [DF], length: 52)
10.89.147.50.41805 > 10.89.147.31.22: . [tcp sum ok] 1:1(0) ack 48 win 11704
<nop,nop,timestamp 226014949 44085169>
03:43:05.693165 IP (tos 0x10, ttl  64, id 53736, offset 0, flags [DF], length: 52)
10.89.147.50.41805 > 10.89.147.31.22: . [tcp sum ok] 1:1(0) ack 160 win 11704
<nop,nop,timestamp 226014949 44085169>
03:43:05.693351 IP (tos 0x10, ttl  64, id 55462, offset 0, flags [DF], length: 316)
10.89.147.31.22 > 10.89.147.50.41805: P [tcp sum ok] 160:424(264) ack 1 win 8576
<nop,nop,timestamp 44085169 226014949>
03:43:05.693493 IP (tos 0x10, ttl  64, id 55463, offset 0, flags [DF], length: 292)
10.89.147.31.22 > 10.89.147.50.41805: P [tcp sum ok] 424:664(240) ack 1 win 8576
<nop,nop,timestamp 44085169 226014949>
03:43:05.693612 IP (tos 0x10, ttl  64, id 55464, offset 0, flags [DF], length: 292)
10.89.147.31.22 > 10.89.147.50.41805: P [tcp sum ok] 664:904(240) ack 1 win 8576
<nop,nop,timestamp 44085169 226014949>
03:43:05.693628 IP (tos 0x10, ttl  64, id 53737, offset 0, flags [DF], length: 52)
10.89.147.50.41805 > 10.89.147.31.22: . [tcp sum ok] 1:1(0) ack 424 win 11704
<nop,nop,timestamp 226014949 44085169>
03:43:05.693654 IP (tos 0x10, ttl  64, id 53738, offset 0, flags [DF], length: 52)
10.89.147.50.41805 > 10.89.147.31.22: . [tcp sum ok] 1:1(0) ack 664 win 11704
<nop,nop,timestamp 226014949 44085169>
03:43:05.693926 IP (tos 0x10, ttl  64, id 55465, offset 0, flags [DF], length: 292)
10.89.147.31.22 > 10.89.147.50.41805: P [tcp sum ok] 904:1144(240) ack 1 win 8576
<nop,nop,timestamp 44085169 226014949>
03:43:05.694043 IP (tos 0x10, ttl  64, id 55466, offset 0, flags [DF], length: 292)
10.89.147.31.22 > 10.89.147.50.41805: P [tcp sum ok] 1144:1384(240) ack 1 win 8576
<nop,nop,timestamp 44085169 226014949>
03:43:05.694163 IP (tos 0x10, ttl  64, id 55467, offset 0, flags [DF], length: 292)
10.89.147.31.22 > 10.89.147.50.41805: P [tcp sum ok] 1384:1624(240) ack 1 win 8576
<nop,nop,timestamp 44085169 226014949>
03:43:05.694209 IP (tos 0x10, ttl  64, id 53739, offset 0, flags [DF], length: 52)
10.89.147.50.41805 > 10.89.147.31.22: . [tcp sum ok] 1:1(0) ack 1384 win 11704
<nop,nop,timestamp 226014950 44085169>
03:43:05.694283 IP (tos 0x10, ttl  64, id 55468, offset 0, flags [DF], length: 292)
10.89.147.31.22 > 10.89.147.50.41805: P [tcp sum ok] 1624:1864(240) ack 1 win 8576
<nop,nop,timestamp 44085169 226014950>
03:43:05.694402 IP (tos 0x10, ttl  64, id 55469, offset 0, flags [DF], length: 292)
10.89.147.31.22 > 10.89.147.50.41805: P [tcp sum ok] 1864:2104(240) ack 1 win 8576
<nop,nop,timestamp 44085169 226014950>
03:43:05.694521 IP (tos 0x10, ttl  64, id 55470, offset 0, flags [DF], length: 292)
10.89.147.31.22 > 10.89.147.50.41805: P [tcp sum ok] 2104:2344(240) ack 1 win 8576
<nop,nop,timestamp 44085169 226014950>
```

```
03:43:05.694690 IP (tos 0x10, ttl  64, id 53740, offset 0, flags [DF], length: 52)
10.89.147.50.41805 > 10.89.147.31.22: . [tcp sum ok] 1:1(0) ack 2344 win 11704
<nop,nop,timestamp 226014950 44085169>
03:43:05.694808 IP (tos 0x10, ttl  64, id 55471, offset 0, flags [DF], length: 300)
10.89.147.31.22 > 10.89.147.50.41805: P [tcp sum ok] 2344:2592(248) ack 1 win 8576
<nop,nop,timestamp 44085169 226014950>
```

Step 3 You can use the **expression** option to limit what you display, for example, only TCP packets.

Note As described in the TCPDUMP man page, the protocol identifiers tcp, udp, and icmp are also keywords and must be escaped by using two back slashes (\\).

```
sensor# packet display GigabitEthernet0/1 verbose expression ip proto \\tcp
Warning: This command will cause significant performance degradation
tcpdump: listening on ge0_1, link-type EN10MB (Ethernet), capture size 65535 bytes
03:42:02.509738 IP (tos 0x10, ttl  64, id 27743, offset 0, flags [DF], length: 88)
10.89.147.31.22 > 64.101.182.54.47039: P [tcp sum ok] 3449098782:3449098830(48) ack
3009767154 win 8704
03:42:02.509834 IP (tos 0x10, ttl  64, id 27744, offset 0, flags [DF], length: 152)
10.89.147.31.22 > 64.101.182.54.47039: P [tcp sum ok] 48:160(112) ack 1 win 8704
03:42:02.510248 IP (tos 0x0, ttl 252, id 55922, offset 0, flags [none], length: 40)
64.101.182.54.47039 > 10.89.147.31.22: . [tcp sum ok] 1:1(0) ack 160 win 8760
03:42:02.511262 IP (tos 0x10, ttl  64, id 27745, offset 0, flags [DF], length: 264)
10.89.147.31.22 > 64.101.182.54.47039: P [tcp sum ok] 160:384(224) ack 1 win 8704
03:42:02.511408 IP (tos 0x10, ttl  64, id 27746, offset 0, flags [DF], length: 248)
10.89.147.31.22 > 64.101.182.54.47039: P [tcp sum ok] 384:592(208) ack 1 win 8704
03:42:02.511545 IP (tos 0x10, ttl  64, id 27747, offset 0, flags [DF], length: 240)
10.89.147.31.22 > 64.101.182.54.47039: P [tcp sum ok] 592:792(200) ack 1 win 8704
```

Step 4 To display information about the packet file:

```
sensor# packet display file-info
Captured by: cisco:25579, Cmd: packet capture GigabitEthernet0/1
Start: 2003/02/03 02:56:48 UTC, End: 2003/02/03 02:56:51 UTC
sensor#
```

Capturing Live Traffic on an Interface

Use the **packet capture** *interface_name* [**snaplen** *length*] [**count** *count*] [**expression** *expression*] command to capture live traffic on an interface.

Only one user can use the **packet capture** command at a time. A second user request results in an error message containing information about the user currently executing the capture.

Caution Executing the **packet capture** command causes significant performance degradation.

The **packet capture** command captures the libpcap output into a local file.

Use the **packet display packet-file** [**verbose**] [**expression** *expression*] command to view the local file.

Use the **packet display file-info** to display information about the local file, if any.

The following options apply:

- *interface_name*—Logical interface name. You can only use an interface name that exists in the system.
- **snaplen**—Maximum number of bytes captured for each packet (optional). The valid range is 68 to 1600. The default is 0.
- **count**—Maximum number of packets to capture (optional). The valid range is 1 to 10000.

  Note If you do not specify this option, the capture terminates after the maximum file size is captured.

- **expression**—Packet-capture filter expression. This expression is passed directly to TCPDUMP and must meet the TCPDUMP expression syntax.

  Note The expression syntax is described in the Wireshark man page.

- **file-info**—Displays information about the stored packet file.

  File-info displays the following information:

  Captured by: *user:id*, Cmd: *cliCmd*

  Start: yyyy/mm/dd hh:mm:ss zone, End: yyyy/mm/dd hh:mm:ss zone or in-progress

  Where

  user = username of user initiating capture

  id = CLI ID of the user

  cliCmd = command entered to perform the capture

- **verbose**—Displays the protocol tree for each packet rather than a one-line summary. This parameter is optional.

To configure the sensor to capture live traffic on an interface, follow these steps:

Step 1 Log in to the sensor using an account with administrator or operator privileges.

Step 2 Capture the live traffic on the interface you are interested in, for example, GigabitEthernet0/1:

```
sensor# packet capture GigabitEthernet0/1
Warning: This command will cause significant performance degradation
tcpdump: WARNING: ge0_1: no IPv4 address assigned
tcpdump: listening on ge0_1, link-type EN10MB (Ethernet), capture size 65535 bytes
125 packets captured
126 packets received by filter
0 packets dropped by kernel
```

Step 3 To view the captured packet file:

```
sensor# packet display packet-file
reading from file /usr/cids/idsRoot/var/packet-file, link-type EN10MB (Ethernet)
03:03:13.216768 802.1d config TOP_CHANGE 8000.00:04:9a:66:35:01.8025 root 8000.0
0:04:6d:f9:e8:82 pathcost 8 age 2 max 20 hello 2 fdelay 15
03:03:13.232881 IP 64.101.182.244.1978 > 10.89.130.108.23: . ack 3266153791 win
64328
03:03:13.232895 IP 10.89.130.108.23 > 64.101.182.244.1978: P 1:157(156) ack 0 wi
n 5840
03:03:13.433136 IP 64.101.182.244.1978 > 10.89.130.108.23: . ack 157 win 65535
```

```
03:03:13.518335 IP 10.89.130.134.42342 > 255.255.255.255.42342: UDP, length: 76
03:03:15.218814 802.1d config TOP_CHANGE 8000.00:04:9a:66:35:01.8025 root 8000.0
0:04:6d:f9:e8:82 pathcost 8 age 2 max 20 hello 2 fdelay 15
03:03:15.546866 IP 64.101.182.244.1978 > 10.89.130.108.23: P 0:2(2) ack 157 win
65535
03:03:15.546923 IP 10.89.130.108.23 > 64.101.182.244.1978: P 157:159(2) ack 2 wi
n 5840
03:03:15.736377 IP 64.101.182.244.1978 > 10.89.130.108.23: . ack 159 win 65533
03:03:17.219612 802.1d config TOP_CHANGE 8000.00:04:9a:66:35:01.8025 root 8000.0
0:04:6d:f9:e8:82 pathcost 8 age 2 max 20 hello 2 fdelay 15
03:03:19.218535 802.1d config TOP_CHANGE 8000.00:04:9a:66:35:01.8025 root 8000.0
0:04:6d:f9:e8:82 pathcost 8 age 2 max 20 hello 2 fdelay 15
03:03:19.843658 IP 64.101.182.143.3262 > 10.89.130.23.445: P 3749577803:37495778
56(53) ack 3040953472 win 64407
03:03:20.174835 IP 161.44.55.250.1720 > 10.89.130.60.445: S 3147454533:314745453
3(0) win 65520 <mss 1260,nop,nop,sackOK>
03:03:21.219958 802.1d config TOP_CHANGE 8000.00:04:9a:66:35:01.8025 root 8000.0
0:04:6d:f9:e8:82 pathcost 8 age 2 max 20 hello 2 fdelay 15
03:03:21.508907 IP 161.44.55.250.1809 > 10.89.130.61.445: S 3152179859:315217985
9(0) win 65520 <mss 1260,nop,nop,sackOK>
03:03:23.221004 802.1d config TOP_CHANGE 8000.00:04:9a:66:35:01.8025 root 8000.0
0:04:6d:f9:e8:82 pathcost 8 age 2 max 20 hello 2 fdelay 15
03:03:23.688099 IP 161.44.55.250.1975 > 10.89.130.63.445: S 3160484670:316048467
0(0) win 65520 <mss 1260,nop,nop,sackOK>
03:03:25.219054 802.1d config TOP_CHANGE 8000.00:04:9a:66:35:01.8025 root 8000.0
0:04:6d:f9:e8:82 pathcost 8 age 2 max 20 hello 2 fdelay 15
03:03:25.846552 IP 172.20.12.10.2984 > 10.89.130.127.445: S 1345848756:134584875
6(0) win 64240 <mss 1460,nop,nop,sackOK>
03:03:26.195342 IP 161.44.55.250.2178 > 10.89.130.65.445: S 3170518052:317051805
2(0) win 65520 <mss 1260,nop,nop,sackOK>
03:03:27.222725 802.1d config TOP_CHANGE 8000.00:04:9a:66:35:01.8025 root 8000.0
0:04:6d:f9:e8:82 pathcost 8 age 2 max 20 hello 2 fdelay 15
03:03:27.299178 IP 161.44.55.250.2269 > 10.89.130.66.445: S 3174717959:317471795
9(0) win 65520 <mss 1260,nop,nop,sackOK>
03:03:27.308798 arp who-has 161.44.55.250 tell 10.89.130.66
03:03:28.383028 IP 161.44.55.250.2349 > 10.89.130.67.445: S 3178636061:317863606
1(0) win 65520 <mss 1260,nop,nop,sackOK>
--MORE--
```

Step 4 To view any information about the packet file:

```
sensor# packet display file-info
Captured by: cisco:8874, Cmd: packet capture GigabitEthernet0/1
Start: 2003/01/07 00:12:50 UTC, End: 2003/01/07 00:15:30 UTC
sensor#
```

Copying the Packet File

Use the **copy packet-file** *destination_url* command to copy the packet file to an FTP or SCP server for saving or further analysis with another tool, such as Wireshark or TCPDUMP.

The following options apply:

- **packet-file**—Locally stored libpcap file that you captured using the **packet capture** command.
- *destination_url*—The location of the destination file to be copied. It can be a URL or a keyword.

Note The exact format of the source and destination URLs varies according to the file.

- ftp:—Destination URL for an FTP network server. The syntax for this prefix is:
  ftp:[//[username@] location]/relativeDirectory]/filename
  ftp:[//[username@]location]//absoluteDirectory]/filename
- scp:—Destination URL for the SCP network server. The syntax for this prefix is:
  scp:[//[username@] location]/relativeDirectory]/filename
  scp:[//[username@] location]//absoluteDirectory]/filename

Note When you use FTP or SCP protocol, you are prompted for a password.

To copy packets files to an FTP or SCP server, follow these steps:

Step 1 Log in to the CLI using an account with administrator privileges.

Step 2 Copy the packet-file to an FTP or SCP server:

```
sensor# copy packet-file scp://jbrown@64.101.182.20/work/
Password: *********
packet-file               100% 1670     0.0KB/s   00:00
sensor#
```

Step 3 View the packet file with Wireshark or TCPDUMP.

Erasing the Packet File

Use the **erase packet-file** command to erase the packet file. There is only one packet file. It is 16 MB and is over-written each time you use the **packet capture** command.

To erase the packet file, follow these steps:

Step 1 Display information about the current captured packet file:

```
sensor# packet display file-info
Captured by: cisco:1514, Cmd: packet capture GigabitEthernet0/1
Start: 2005/02/15 03:55:00 CST, End: 2005/02/15 03:55:05 CST
sensor#
```

Step 2 Erase the packet file::

```
sensor# erase packet-file
sensor#
```

Step 3 Verify that you have erased the packet file:

```
sensor# packet display file-info
No packet-file available.
sensor#
```

CHAPTER 10

Configuring Attack Response Controller for Blocking and Rate Limiting

This chapter provides information for setting up Attack Response Controller (ARC) to perform blocking and rate limiting on the sensor.

Note ARC was formerly known as Network Access Controller. The name has been changed for IPS 5.1, although the CLI still contains the terms **nac** and **network-access**.

This contains the following sections:

Understanding Blocking

ARC, the blocking application on the sensor, starts and stops blocks on routers, switches, PIX Firewalls, FWSM, and ASA. ARC blocks the IP address on the devices it is managing. It sends the same block to all the devices it is managing, including any other master blocking sensors. ARC monitors the time for the block and removes the block after the time has expired.

Caution If FWSM is configured in multi-mode, blocking is not supported for the admin context. Blocking is only supported in single mode and in multi-mode customer context.

Note ARC was designed to complete the action response for a new block in no more than 4 to 7 seconds. In most cases, it completes the action response in less time. To meet this performance goal, you should not configure the sensor to perform blocks at too high a rate or to manage too many blocking devices and interfaces. We recommend that the maximum number of blocks not exceed 250 and the maximum number of blocking items not exceed 10. To calculate the maximum number of blocking items, a firewall, ASA, or FWSM counts as one blocking item per blocking context. A router counts as one blocking item per blocking interface/direction. A switch running Catalyst software counts as one blocking item per blocking VLAN. If the recommended limits are exceeded, ARC may not apply blocks in a timely manner or may not be able to apply blocks at all.

For firewalls, such as ASA, PIX Firewall 7.0, and FWSM 2.1 or greater, configured in multi-mode, IPS 5.1 does not include VLAN information in the block request. Therefore you must make sure the IP addresses being blocked are correct for each firewall. For example, the sensor is monitoring packets on a firewall customer context that is configured for VLAN A, but is blocking on a different firewall customer context that is configured for VLAN B. Addresses that trigger blocks on VLAN A may refer to a different host on VLAN B.

There are three types of blocks:

- Host block—Blocks all traffic from a given IP address.
- Connection block—Blocks traffic from a given source IP address to a given destination IP address and destination port.

  Multiple connection blocks from the same source IP address to either a different destination IP address or destination port automatically switch the block from a connection block to a host block.

  Note Connection blocks are not supported on firewalls. Firewalls only support host blocks with additional connection information.

- Network block—Blocks all traffic from a given network.

  You can initiate host and connection blocks manually or automatically when a signature is triggered. You can only initiate network blocks manually.

Caution Do not confuse blocking with the sensor's ability to drop packets. The sensor can drop packets when the following actions are configured for a sensor in inline mode: deny packet inline, deny connection inline, and deny attacker inline.

For automatic blocks, you must select **Request Block Host** or **Request Block Connection** as the event action for particular signatures, and add them to any event action overrides you have configured, so that SensorApp sends a block request to ARC when the signature is triggered. When ARC receives the block request from SensorApp, it updates the device configurations to block the host or connection. For the procedure to add the Request Block Host or Request Block Connection event actions to the signature, see Assigning Actions to Signatures, page 7-11. Or, for the procedure for configuring overrides that add the Request Block Host or Request Block Connection event actions to alerts of specific risk ratings, see Configuring Event Action Overrides, page 6-11.

On Cisco routers and Catalyst 6500 series switches, ARC creates blocks by applying ACLs or VACLs. ACLs and VACLs permit or deny passage of data packets through interface directions or VLANs. Each ACL or VACL contains permit and deny conditions that apply to IP addresses. The PIX Firewall, FWSM, and ASA do not use ACLs or VACLs. The built-in **shun** and **no shun** command is used.

Caution The ACLs that ARC makes should never be modified by you or any other system. These ACLs are temporary and new ACLs are constantly being created by the sensor. The only modifications that you can make are to the Pre- and Post-Block ACLs. For more information, see How the Sensor Manages Devices, page 10-21.

You need the following information for ARC to manage a device:

- Login user ID (if the device is configured with AAA)
- Login password
- Enable password (not needed if the user has enable privileges)
- Interfaces to be managed (for example, ethernet0, vlan100)
- Any existing ACL or VACL information you want applied at the beginning (Pre-Block ACL or VACL) or end (Post-Block ACL or VACL) of the ACL or VACL that will be created

  This does not apply to a PIX firewall, FWSM, or ASA because they do not use ACLs to block.
- Whether you are using Telnet or SSH to communicate with the device
- IP addresses (host or range of hosts) you never want blocked
- How long you want the blocks to last

Tip To check the status of ARC, enter **show statistics network-access** at the `sensor#`. The output shows the devices you are managing, any active blocks and rate limits, and the status of all devices. Or in the IDM, choose **Monitoring > Statistics** to see the status of ARC.

Note ARC is formerly known as Network Access Controller. Although the name has been changed, IDM and the CLI contain references to Network Access Controller, **nac**, and **network-access**.

Understanding Rate Limiting

Rate limiting lets sensors restrict the rate of specified traffic classes on network devices. Rate limit responses are supported for the Host Flood and Net Flood engines, and the TCP half-open SYN signature. ARC can configure rate limits on network devices running Cisco IOS version 12.3 or later. For configuring rate limiting on routers, see Configuring Blocking and Rate Limiting Devices, page 10-20. Master blocking sensors can also forward rate limit requests to blocking forwarding sensors. For more information, see Configuring the Sensor to be a Master Blocking Sensor, page 10-28.

Tip To check the status of ARC, enter **show statistics network-access** at the `sensor#`. The output shows the devices you are managing, any active blocks and rate limits, and the status of all devices. Or in IDM, choose **Monitoring > Statistics** to see the status of ARC.

To add a rate limit, you specify a combination of protocol, destination IP address, and data value to match one of the signatures that are allowed to generate rate limit events.You must also set the action to Request Rate Limit and set the percentage for these signatures.

Table 10-1 lists the supported signatures and parameters.

Table 10-1 Rate Limiting Signatures

Signature ID	Signature Name	Protocol	Destination IP Address Allowed	Data
2152	ICMP Flood Host	ICMP	Yes	echo-request
2153	ICMP Smurf Attack	ICMP	Yes	echo-reply
4002	UDP Flood Host	UDP	Yes	none
6901	Net Flood ICMP Reply	ICMP	No	echo-reply
6902	Net Flood ICMP Request	ICMP	No	echo-request
6903	Net Flood ICMP Any	ICMP	No	None
6910	Net Flood UDP	UDP	No	None
6920	Net Flood TCP	TCP	No	None
3050	TCP HalfOpenSyn	TCP	No	halfOpenSyn

Before Configuring Attack Response Controller

Before you configure ARC for blocking or rate limiting, make sure you do the following:

- Analyze your network topology to understand which devices should be blocked by which sensor, and which addresses should never be blocked.

Caution Two sensors cannot control blocking or rate limiting on the same device. If this situation is needed, configure one sensor as the master blocking sensor to manage the devices and the other sensors can forward their requests to the master blocking sensor. For the procedure, see Configuring the Sensor to be a Master Blocking Sensor, page 10-28.

Note When you add a master blocking sensor, you reduce the number of blocking devices per sensor. For example, if you want to block on 10 firewalls and 10 routers with one blocking interface/direction each, you can assign 10 to the sensor and assign the other 10 to a master blocking sensor.

- Gather the usernames, device passwords, enable passwords, and connections types (Telnet or SSH) needed to log in to each device.
- Know the interface names on the devices.
- Know the names of the Pre-Block ACL or VACL and the Post-Block ACL or VACL if needed.
- Understand which interfaces should and should not be blocked and in which direction (in or out). You do not want to accidentally shut down an entire network.

Supported Devices

By default, ARC supports up to 250 devices in any combination. The following devices are supported for blocking by ARC:

Caution If the recommended limits are exceeded, ARC may not apply blocks in a timely manner or may not be able to apply blocks at all.

- Cisco series routers using Cisco IOS 11.2 or later (ACLs):
  - Cisco 1600 series router
  - Cisco 1700 series router
  - Cisco 2500 series router
  - Cisco 2600 series router
  - Cisco 2800 series router
  - Cisco 3600 series router
  - Cisco 3800 series router
  - Cisco 7200 series router
  - Cisco 7500 series router
- Catalyst 5000 switches with RSM with IOS 11.2(9)P or later (ACLs)
- Catalyst 6500 switches and 7600 routers with IOS 12.1(13)E or later (ACLs)
- Catalyst 6500 switches 7600 routers with Catalyst software version 7.5(1) or later (VACLs)
  - Supervisor Engine 1A with PFC
  - Supervisor Engine 1A with MSFC1
  - Supervisor Engine 1A with MFSC2
  - Supervisor Engine 2 with MSFC2
  - Supervisor Engine 720 with MSFC3

  Note We support VACL blocking on the Supervisor Engine and ACL blocking on the MSFC.

- PIX Firewall with version 6.0 or later (**shun** command)
  - 501
  - 506E
  - 515E
  - 525
  - 535

- ASA with version 7.0 or later (**shun** command)
  - ASA-5510
  - ASA-5520
  - ASA-5540
- FWSM 1.1 or later (**shun** command)

You configure blocking using either ACLs, VACLS, or the **shun** command. All firewall and ASA models support the **shun** command.

The following devices are supported for rate limiting by ARC:

- Cisco series routers using Cisco IOS 12.3 or later:
  - Cisco 1700 series router
  - Cisco 2500 series router
  - Cisco 2600 series router
  - Cisco 2800 series router
  - Cisco 3600 series router
  - Cisco 3800 series router
  - Cisco 7200 series router
  - Cisco 7500 series router

Caution ARC cannot perform rate limits on 7500 routers with VIP. ARC reports the error but cannot rate limit.

Configuring Blocking Properties

You can change the default blocking properties. It is best to use the default properties, but if you need to change them, use the following procedures:

- Allowing the Sensor to Block Itself, page 10-7
- Disabling Blocking, page 10-8
- Setting Maximum Block Entries, page 10-10
- Setting the Block Time, page 10-12
- Enabling ACL Logging, page 10-13
- Enabling Writing to NVRAM, page 10-14
- Logging All Blocking Events and Errors, page 10-15
- Configuring the Maximum Number of Blocking Interfaces, page 10-16
- Configuring Addresses Never to Block, page 10-18

Allowing the Sensor to Block Itself

Use the **allow-sensor-block {true | false}** command in the service network-access submode to configure the sensor to block itself.

Caution We recommend that you do not permit the sensor to block itself, because it may stop communicating with the blocking device. You can configure this option if you can ensure that if the sensor creates a rule to block its own IP address, it will not prevent the sensor from accessing the blocking device.

To allow the sensor to block itself, follow these steps:

Step 1 Log in to the CLI using an account with administrator privileges.

Step 2 Enter network access submode:

```
sensor# configure terminal
sensor(config)# service network-access
```

Step 3 Enter general submode:

```
sensor(config-net)# general
```

Step 4 Configure the sensor to block itself:

```
sensor(config-net-gen)# allow-sensor-block true
```

By default, this value is **false**.

Step 5 Verify the settings:

```
sensor(config-net-gen)# show settings
   general
   ----------------------------------------------
      log-all-block-events-and-errors: true <defaulted>
      enable-nvram-write: false <defaulted>
      enable-acl-logging: false <defaulted>
      allow-sensor-block: true default: false
      block-enable: true default: true
      block-max-entries: 100 default: 250
      max-interfaces: 250 <defaulted>
      master-blocking-sensors (min: 0, max: 100, current: 0)
      ----------------------------------------------
      ----------------------------------------------
      never-block-hosts (min: 0, max: 250, current: 1)
      ----------------------------------------------
         ip-address: 11.11.11.11
         ----------------------------------------------
      ----------------------------------------------
      never-block-networks (min: 0, max: 250, current: 1)
      ----------------------------------------------
         ip-address: 12.12.0.0/16
         ----------------------------------------------
      ----------------------------------------------
      block-hosts (min: 0, max: 250, current: 0)
      ----------------------------------------------
--MORE--
```

Step 6 Configure the sensor not to block itself:

```
sensor(config-net-gen)# allow-sensor-block false
```

Step 7 Verify the setting:

```
sensor(config-net-gen)# show settings
   general
   ----------------------------------------------
      log-all-block-events-and-errors: true <defaulted>
      enable-nvram-write: false <defaulted>
      enable-acl-logging: false <defaulted>
      allow-sensor-block: false default: false
      block-enable: true default: true
      block-max-entries: 100 default: 250
      max-interfaces: 250 <defaulted>
      master-blocking-sensors (min: 0, max: 100, current: 0)
      ----------------------------------------------
      ----------------------------------------------
      never-block-hosts (min: 0, max: 250, current: 1)
      ----------------------------------------------
         ip-address: 11.11.11.11
         ----------------------------------------------
      ----------------------------------------------
      never-block-networks (min: 0, max: 250, current: 1)
      ----------------------------------------------
         ip-address: 12.12.0.0/16
         ----------------------------------------------
      ----------------------------------------------
      block-hosts (min: 0, max: 250, current: 0)
      ----------------------------------------------
--MORE--
```

Step 8 Exit network access submode:

```
sensor(config-net-gen)# exit
sensor(config-net)# exit
Apply Changes:?[yes]:
```

Step 9 Press **Enter** to apply the changes or enter **no** to discard them.

Disabling Blocking

Use the **block-enable {true | false}** command in the service network access submode to enable or disable blocking on the sensor.

Note For blocking to operate, you must set up devices to do the blocking. For the procedures, see Configuring the Sensor to Manage Cisco Routers, page 10-22, and Configuring the Sensor to Manage Catalyst 6500 Series Switches and Cisco 7600 Series Routers, page 10-25.

By default, blocking is enabled on the sensor. If ARC is managing a device and you need to manually configure something on that device, you should disable blocking first. You want to avoid a situation in which both you and ARC could be making a change at the same time on the same device. This could cause the device and/or ARC to crash.

Caution If you disable blocking for maintenance on the devices, make sure you enable it after the maintenance is complete or the network will be vulnerable to attacks that would otherwise be blocked

Note While blocking is disabled, ARC continues to receive blocks and track the time on active blocks, but will not apply new blocks or remove blocks from the managed devices. After blocking is reenabled, the blocks on the devices are updated.

To disable blocking or rate limiting, follow these steps:

Step 1 Log in to the CLI using an account with administrator privileges.

Step 2 Enter network access submode:

```
sensor# configure terminal
sensor(config)# service network-access
sensor(config-net)#
```

Step 3 Enter general submode:

```
sensor(config-net)# general
```

Step 4 Disable blocking on the sensor:

```
sensor(config-net-gen)# block-enable false
```

By default, this value is **true**.

Step 5 Verify the settings:

```
sensor(config-net-gen)# show settings
   general
   ----------------------------------------------
      log-all-block-events-and-errors: true <defaulted>
      enable-nvram-write: false <defaulted>
      enable-acl-logging: false <defaulted>
      allow-sensor-block: false default: false
      block-enable: false default: true
      block-max-entries: 100 default: 250
      max-interfaces: 250 <defaulted>
      master-blocking-sensors (min: 0, max: 100, current: 0)
      ----------------------------------------------
      ----------------------------------------------
      never-block-hosts (min: 0, max: 250, current: 1)
      ----------------------------------------------
         ip-address: 11.11.11.11
         ----------------------------------------------
      ----------------------------------------------
      never-block-networks (min: 0, max: 250, current: 1)
      ----------------------------------------------
         ip-address: 12.12.0.0/16
         ----------------------------------------------
      ----------------------------------------------
      block-hosts (min: 0, max: 250, current: 0)
      ----------------------------------------------
--MORE--
```

Step 6 Enable blocking on the sensor:

```
sensor(config-net-gen)# block-enable true
```

Step 7 Verify that the setting has been returned to the default:

```
sensor(config-net-gen)# show settings
   general
   ----------------------------------------------
```

```
      log-all-block-events-and-errors: true <defaulted>
      enable-nvram-write: false <defaulted>
      enable-acl-logging: false <defaulted>
      allow-sensor-block: false default: false
      block-enable: true default: true
      block-max-entries: 100 default: 250
      max-interfaces: 250 <defaulted>
      master-blocking-sensors (min: 0, max: 100, current: 0)
      ----------------------------------------------
      ----------------------------------------------
      never-block-hosts (min: 0, max: 250, current: 1)
      ----------------------------------------------
         ip-address: 11.11.11.11
         ----------------------------------------------
      ----------------------------------------------
      never-block-networks (min: 0, max: 250, current: 1)
      ----------------------------------------------
         ip-address: 12.12.0.0/16
         ----------------------------------------------
      ----------------------------------------------
      block-hosts (min: 0, max: 250, current: 0)
      ----------------------------------------------
--MORE--
```

Step 8 Exit network access submode:

```
sensor(config-net-gen)# exit
sensor(config-net)# exit
Apply Changes:?[yes]:
```

Step 9 Press **Enter** to apply the changes or enter **no** to discard them.

Setting Maximum Block Entries

Use the **block-max-entries** command in the service network access submode to configure the maximum block entries.

You can set how many blocks are to be maintained simultaneously (1 to 65535). The default value is 250.

Caution We do not recommend setting the maximum block entries higher than 250. Some devices have problems with larger numbers of ACL or shun entries. Refer to the documentation for each device to determine its limits before increasing this number.

Note The number of blocks will not exceed the maximum block entries. If the maximum is reached, new blocks will not occur until existing blocks time out and are removed.

To change the maximum number of block entries, follow these steps:

Step 1 Log in to the CLI using an account with administrator privileges.

Step 2 Enter network access submode:

```
sensor# configure terminal
sensor(config)# service network-access
```

```
sensor(config-net)#
```

Step 3 Enter general submode:

```
sensor(config-net)# general
```

Step 4 Change the maximum number of block entries:

```
sensor(config-net-gen)# block-max-entries 100
```

Step 5 Verify the setting:

```
sensor(config-net-gen)# show settings
   general
   ----------------------------------------------
      log-all-block-events-and-errors: true <defaulted>
      enable-nvram-write: false <defaulted>
      enable-acl-logging: false <defaulted>
      allow-sensor-block: false default: false
      block-enable: true <defaulted>
      block-max-entries: 100 default: 250
      max-interfaces: 250 <defaulted>
      master-blocking-sensors (min: 0, max: 100, current: 0)
      ----------------------------------------------
      ----------------------------------------------
      never-block-hosts (min: 0, max: 250, current: 1)
      ----------------------------------------------
         ip-address: 11.11.11.11
         ----------------------------------------------
      ----------------------------------------------
      never-block-networks (min: 0, max: 250, current: 1)
      ----------------------------------------------
         ip-address: 12.12.0.0/16
         ----------------------------------------------
      ----------------------------------------------
      block-hosts (min: 0, max: 250, current: 0)
      ----------------------------------------------
--MORE--
```

Step 6 To return to the default value of 250 blocks:

```
sensor(config-net-gen)# default block-max-entries
```

Step 7 Verify the setting:

```
sensor(config-net-gen)# show settings
   general
   ----------------------------------------------
      log-all-block-events-and-errors: true <defaulted>
      enable-nvram-write: false <defaulted>
      enable-acl-logging: false <defaulted>
      allow-sensor-block: false default: false
      block-enable: true <defaulted>
      block-max-entries: 250 <defaulted>
      max-interfaces: 250 <defaulted>
      master-blocking-sensors (min: 0, max: 100, current: 0
      ----------------------------------------------
      ----------------------------------------------
      never-block-hosts (min: 0, max: 250, current: 1)
      ----------------------------------------------
         ip-address: 11.11.11.11
         ----------------------------------------------
      ----------------------------------------------
      never-block-networks (min: 0, max: 250, current: 1)
      ----------------------------------------------
```

```
         ip-address: 12.12.0.0/16
         ----------------------------------------------
      ---------------------------------------------
      block-hosts (min: 0, max: 250, current: 0)
      ---------------------------------------------
--MORE--
```

Step 8 Exit network access submodule:

```
sensor(config-net-gen)# exit
sensor(config-net)# exit
Apply Changes:?[yes]:
```

Step 9 Press **Enter** to apply the changes or enter **no** to discard them.

Setting the Block Time

Use the **global-block-timeout** command in the service event action rules submode to change the amount of time an automatic block lasts. The default is 30 minutes.

Note If you change the default block time, you are changing a signature parameter, which affects all signatures.

Note The time for manual blocks is set when you request the block.

To change the default block time, follow these steps:

Step 1 Log in to the CLI using an account with administrator privileges.

Step 2 Enter event action rules submode:

```
sensor# configure terminal
sensor(config)# service event-action-rules rules0
sensor(config-rul)#
```

Step 3 Enter general submode:

```
sensor(config-rul)# general
```

Step 4 Configure the block time:

```
sensor(config-rul-gen)# global-block-timeout 60
```

The value is the time duration of the block event in minutes (0 to 10000000).

Step 5 Verify the setting:

```
sensor(config-rul-gen)# show settings
   general
   ----------------------------------------------
      global-overrides-status: Enabled <defaulted>
      global-filters-status: Enabled <defaulted>
      global-summarization-status: Enabled <defaulted>
      global-metaevent-status: Enabled <defaulted>
      global-deny-timeout: 3600 <defaulted>
      global-block-timeout: 60 default: 30
```

```
      max-denied-attackers: 10000 <defaulted>
   ----------------------------------------------
sensor(config-rul-gen)#
```

Step 6 Exit event action rules submode:

```
sensor(config-rul-gen)# exit
sensor(config-rul)# exit
Apply Changes:?[yes]:
```

Step 7 Press **Enter** to apply the changes or enter **no** to discard them.

Note There is a time delay while the signatures are updated.

Enabling ACL Logging

Use the **enable-acl-logging {true | false}** command in the service network access submode to enable ACL logging, which causes ARC to append the log parameter to block entries in the ACL or VACL. This causes the device to generate syslog events when packets are filtered. Enable ACL logging only applies to routers and switches. The default is disabled.

To enable ACL logging, follow these steps:

Step 1 Log in to the CLI using an account with administrator privileges.

Step 2 Enter network access submode:

```
sensor# configure terminal
sensor(config)# service network-access
sensor(config-net)#
```

Step 3 Enter general submode:

```
sensor(config-net)# general
```

Step 4 Enable ACL logging:

```
sensor(config-net-gen)# enable-acl-logging true
```

Step 5 Verify that ACL logging is enabled:

```
sensor(config-net-gen)# show settings
   general
   ----------------------------------------------
      log-all-block-events-and-errors: true <defaulted>
      enable-nvram-write: false <defaulted>
      enable-acl-logging: true default: false
      allow-sensor-block: false <defaulted>
      block-enable: true <defaulted>
      block-max-entries: 250 <defaulted>
      max-interfaces: 250 <defaulted>
      master-blocking-sensors (min: 0, max: 100, current: 0)
      ----------------------------------------------
```

Step 6 To disable ACL logging, use the **false** keyword:

```
sensor(config-net-gen)# enable-acl-logging false
```

Step 7 Verify that ACL logging is disabled:

```
sensor(config-net-gen)# show settings
   general
   ----------------------------------------------
      log-all-block-events-and-errors: true <defaulted>
      enable-nvram-write: false <defaulted>
      enable-acl-logging: false default: false
      allow-sensor-block: false <defaulted>
      block-enable: true <defaulted>
      block-max-entries: 250 <defaulted>
      max-interfaces: 250 <defaulted>
      master-blocking-sensors (min: 0, max: 100, current: 0)
      ----------------------------------------------
```

Step 8 Exit network access mode:

```
sensor(config-net-gen)# exit
sensor(config-net)# exit
Apply Changes:?[yes]:
```

Step 9 Press **Enter** to apply the changes or enter **no** to discard them.

Enabling Writing to NVRAM

Use the **enable-nvram-write {true | false}** command to configure the sensor to have the router write to NVRAM when ARC first connects. If enable-nvram-write is enabled, NVRAM is written each time the ACLs are updated. The default is disabled.

Enabling NVRAM writing ensures that all changes for blocking are written to NVRAM. If the router is rebooted, the correct blocks will still be active. If NVRAM writing is disabled, a short time without blocking occurs after a router reboot. And not enabling NVRAM writing increases the life of the NVRAM and decreases the time for new blocks to be configured.

To enable writing to NVRAM, follow these steps:

Step 1 Log in to the CLI using an account with administrator privileges.

Step 2 Enter network access submode:

```
sensor# configure terminal
sensor(config)# service network-access
sensor(config-net)#
```

Step 3 Enter general submode:

```
sensor(config-net)# general
```

Step 4 Enable writing to NVRAM:

```
sensor(config-net-gen)# enable-nvram-write true
```

Step 5 Verify that writing to NVRAM is enabled:

```
sensor(config-net-gen)# show settings
   general
   ----------------------------------------------
      log-all-block-events-and-errors: true <defaulted>
      enable-nvram-write: true default: false
      enable-acl-logging: false default: false
```

```
      allow-sensor-block: false <defaulted>
      block-enable: true <defaulted>
      block-max-entries: 250 <defaulted>
      max-interfaces: 250 <defaulted>
      master-blocking-sensors (min: 0, max: 100, current: 0)
      ----------------------------------------------
```

Step 6 Disable writing to NVRAM:

```
sensor(config-net-gen)# enable-nvram-write false
```

Step 7 Verify that writing to NVRAM is disabled:

```
sensor(config-net-gen)# show settings
   general
   ----------------------------------------------
      log-all-block-events-and-errors: true <defaulted>
      enable-nvram-write: false default: false
      enable-acl-logging: false default: false
      allow-sensor-block: false <defaulted>
      block-enable: true <defaulted>
      block-max-entries: 250 <defaulted>
      max-interfaces: 250 <defaulted>
      master-blocking-sensors (min: 0, max: 100, current: 0)
      ----------------------------------------------
```

Step 8 Exit network access submode:

```
sensor(config-net-gen)# exit
sensor(config-net)# exit
Apply Changes:?[yes]:
```

Step 9 Press **Enter** to apply the changes or enter **no** to discard them.

Logging All Blocking Events and Errors

Use the **log-all-block-events-and-errors {true | false}** command in the service network access submode to configure the sensor to log events that follow blocks from start to finish. For example, when a block is added to or removed from a device, an event is logged. You may not want all these events and errors to be logged. Disabling **log-all-block-events-and-errors** suppresses the new events and errors. The default is enabled.

To disable blocking event and error logging, follow these steps:

Step 1 Log in to the CLI using an account with administrator privileges.

Step 2 Enter network access mode:

```
sensor# configure terminal
sensor(config)# service network-access
sensor(config-net)#
```

Step 3 Enter general submode:

```
sensor(config-net)# general
```

Step 4 Disable blocking event and error logging:

```
sensor(config-net-gen)# log-all-block-events-and-errors false
```

Step 5 Verify that logging is disabled:

```
sensor(config-net-gen)# show settings
   general
   ----------------------------------------------
      log-all-block-events-and-errors: false default: true
      enable-nvram-write: false default: false
      enable-acl-logging: false default: false
      allow-sensor-block: false <defaulted>
      block-enable: true <defaulted>
      block-max-entries: 250 <defaulted>
      max-interfaces: 250 <defaulted>
      master-blocking-sensors (min: 0, max: 100, current: 0)
      ----------------------------------------------
```

Step 6 Enable blocking event and error logging:

```
sensor(config-net-gen)# log-all-block-events-and-errors true
```

Step 7 Verify that logging is enabled:

```
sensor(config-net-gen)# show settings
   general
   ----------------------------------------------
      log-all-block-events-and-errors: true default: true
      enable-nvram-write: false default: false
      enable-acl-logging: false default: false
      allow-sensor-block: false <defaulted>
      block-enable: true <defaulted>
      block-max-entries: 250 <defaulted>
      max-interfaces: 250 <defaulted>
      master-blocking-sensors (min: 0, max: 100, current: 0)
      ----------------------------------------------
```

Step 8 Exit network access mode:

```
sensor(config-net-gen)# exit
sensor(config-net)# exit
Apply Changes:?[yes]:
```

Step 9 Press **Enter** to apply the changes or enter **no** to discard them.

Configuring the Maximum Number of Blocking Interfaces

Use the **max-interfaces** command to configure the maximum number of interfaces for performing blocks. For example, a PIX Firewall counts as one interface. A router with one interface counts as one, but a router with two interfaces counts as two. At most you can configure 250 blocking interfaces on a router, switch, or firewall. You can configure up to 250 Catalyst 6K switches, 250 routers, and 250 firewalls.

The **max-interfaces** command configures the limit of the sum total of all interfaces and devices. In addition to configuring the limit on the sum total of interfaces and devices, there is a fixed limit on the number of blocking interfaces you can configure per device. Use the **show settings** command in network access mode to view the specific maximum limits per device.

To configure the maximum number of blocking interfaces, follow these steps:

Step 1 Log in to the CLI using an account with administrator privileges.

Step 2 Enter network access mode:

```
sensor# configure terminal
sensor(config)# service network-access
sensor(config-net)#
```

Step 3 Enter general submode:

```
sensor(config-net)# general
```

Step 4 Configure the maximum number of interfaces:

```
sensor(config-net-gen)# max-interfaces 50
```

Step 5 Verify the number of maximum interfaces:

```
sensor(config-net-gen)# show settings
   general
   ----------------------------------------------
      log-all-block-events-and-errors: true default: true
      enable-nvram-write: false default: false
      enable-acl-logging: false default: false
      allow-sensor-block: false <defaulted>
      block-enable: true <defaulted>
      block-max-entries: 250 <defaulted>
      max-interfaces: 50 default: 250
      master-blocking-sensors (min: 0, max: 100, current: 0)
      ----------------------------------------------
```

Step 6 Return the setting to the default of 250:

```
sensor(config-net-gen)# default max-interfaces
```

Step 7 Verify the default setting:

```
sensor(config-net-gen)# show settings

   general
   ----------------------------------------------
      log-all-block-events-and-errors: true default: true
      enable-nvram-write: false default: false
      enable-acl-logging: false default: false
      allow-sensor-block: false <defaulted>
      block-enable: true <defaulted>
      block-max-entries: 250 <defaulted>
      max-interfaces: 250 <defaulted>
      master-blocking-sensors (min: 0, max: 100, current: 0)
      ----------------------------------------------
```

Step 8 Exit network access mode:

```
sensor(config-net-gen)# exit
sensor(config-net)# exit
Apply Changes:?[yes]:
```

Step 9 Press **Enter** to apply the changes or enter **no** to discard them.

Configuring Addresses Never to Block

Use the **never-block-hosts** and the **never-block-networks** commands in the service network access submode to configure hosts and network that should never be blocked.

The following options apply:

- *ip_address*—IP address of the device that should never be blocked.
- *ip_address/netmask*— IP address of the network that should never be blocked. The format is A.B.C.D/nn.

You must tune your sensor to identify hosts and networks that should never be blocked, not even manually, because you may have a trusted network device whose normal, expected behavior appears to be an attack. Such a device should never be blocked, and trusted, internal networks should never be blocked.

You can specify a single host or an entire network.

Note The **never-block-hosts** and the **never-block-networks** commands apply only to the Request Block Host and Request Block Connection event actions. It does not apply to the Deny Attacker Inline, Deny Connection Inline, or Deny Packet Inline event actions. Use event action rules to filter out the hosts that you do not want blocked, denied, or dropped. For more information, see Configuring Event Action Filters, page 6-13.

To set up addresses never to be blocked by blocking devices, follow these steps:

Step 1 Log in to the CLI using an account with administrator privileges.

Step 2 Enter network access submode:

```
sensor# configure terminal
sensor(config)# service network-access
sensor(config-net)#
```

Step 3 Enter general submode:

```
sensor(config-net)# general
```

Step 4 Define the address that should never be blocked:

- For a single host:

```
sensor(config-net-gen)# never-block-hosts 10.16.0.0
```

- For an entire network:

```
sensor(config-net-gen)# never-block-networks 10.0.0.0/8
```

Step 5 Verify the settings:

```
sensor(config-net-gen)# show settings
   general
   -----------------------------------------------
      log-all-block-events-and-errors: true <defaulted>
      enable-nvram-write: false <defaulted>
      enable-acl-logging: false <defaulted>
      allow-sensor-block: false default: false
      block-enable: true default: true
      block-max-entries: 100 default: 250
      max-interfaces: 250 <defaulted>
```

```
      master-blocking-sensors (min: 0, max: 100, current: 0)
      ----------------------------------------------
      ----------------------------------------------
      never-block-hosts (min: 0, max: 250, current: 2)
      ----------------------------------------------
         ip-address: 10.16.0.0
         ----------------------------------------------
         ip-address: 11.11.11.11
         ----------------------------------------------
      ----------------------------------------------
      never-block-networks (min: 0, max: 250, current: 2)
      ----------------------------------------------
         ip-address: 10.0.0.0/8
         ----------------------------------------------
         ip-address: 12.12.0.0/16
--MORE--
```

Step 6 Exit network access submode:

```
sensor(config-net-gen)# exit
sensor(config-net)# exit
Apply Changes:?[yes]:
```

Step 7 Press **Enter** to apply the changes or enter **no** to discard them.

Configuring User Profiles

Use the **user-profiles** *profile_name* command in the service network access submode to set up user profiles for the other devices that the senor will manage. The user profiles contain userid, password, and enable password information. For example, routers that all share the same passwords and usernames can be under one user profile.

Note If the username or password is not needed to log in to the device, do not set a value for it.

Note You MUST create a user profile before configuring the blocking device.

To set up user profiles, follow these steps:

Step 1 Log in to the CLI using an account with administrator privileges.

Step 2 Enter network access mode:

```
sensor# configure terminal
sensor(config)# service network-access
sensor(config-net)#
```

Step 3 Create the user profile name:

```
sensor(config-net)# user-profiles PROFILE1
```

Step 4 Type the username for that user profile:

```
sensor(config-net-use)# username username
```

Step 5 Specify the password for the user:

```
sensor(config-net-use)# password
Enter password[]: ********
Re-enter password ********
```

Step 6 Specify the enable password for the user:

```
sensor(config-net-use)# enable-password
Enter enable-password[]: ********
Re-enter enable-password ********
```

Step 7 Verify the settings:

```
sensor(config-net-use)# show settings
   profile-name: PROFILE1
   ----------------------------------------------
      enable-password: <hidden>
      password: <hidden>
      username: jsmith default:
   ----------------------------------------------
sensor(config-net-use)#
```

Step 8 Exit network access submode:

```
sensor(config-net-use)# exit
sensor(config-net)# exit
Apply Changes:?[yes]:
```

Step 9 Press **Enter** to apply the changes or enter **no** to discard them.

Configuring Blocking and Rate Limiting Devices

This section describes how to configure devices that the sensor uses to block or rate limit. It contains the following topics:

How the Sensor Manages Devices

Note ACLs do not apply to rate limiting devices.

ARC uses ACLs on Cisco routers and switches to manage those devices. These ACLs are built as follows:

1. A **permit** line with the sensor's IP address or, if specified, the NAT address of the sensor.

   Note If you permit the sensor to be blocked, this line does not appear in the ACL.

2. Pre-Block ACL (if specified)

   This ACL must already exist on the device.

   Note ARC reads the lines in the ACL and copies these lines to the beginning of the ACL.

3. Any active blocks
4. Either:
   - Post-Block ACL (if specified)

     This ACL must already exist on the device.

     Note ARC reads the lines in the ACL and copies these lines to the end of the ACL.

     Note Make sure the last line in the ACL is **permit ip any any** if you want all unmatched packets to be permitted.

   - **permit ip any any** (not used if a Post-Block ACL is specified)

ARC uses two ACLs to manage devices. Only one is active at any one time. It uses the offline ACL name to build the new ACL, then applies it to the interface. ARC then reverses the process on the next cycle.

Caution The ACLs that ARC makes should never be modified by you or any other system. These ACLs are temporary and new ACLs are constantly being created by the sensor. The only modifications that you can make are to the Pre- and Post-Block ACLs.

Note The ACLs that ARC creates are not removed from the managed device after you configure ARC to no longer manage that device. You must remove the ACLs manually on any device that ARC formerly managed.

If you need to modify the Pre-Block or Post-Block ACL, do the following:

1. Disable blocking on the sensor.
2. Make the changes to the device's configuration.

3. Reenable blocking on the sensor.

When blocking is reenabled, the sensor reads the new device configuration. For the procedure, see Configuring Blocking Properties, page 10-6.

Caution A single sensor can manage multiple devices, but you cannot use multiple sensors to control a single device. In this case, use a master blocking sensor. For the procedure, see Configuring the Sensor to be a Master Blocking Sensor, page 10-28.

Configuring the Sensor to Manage Cisco Routers

This section describes how to configure the sensor to manage Cisco routers. It contains the following topics:

Routers and ACLs

You create and save Pre-Block and Post-Block ACLs in your router configuration. These ACLs must be extended IP ACLs, either named or numbered. See your router documentation for more information on creating ACLs.

Note Pre-Block and Post-Block ACLS do not apply to rate limiting.

Enter the names of these ACLs that are already configured on your router in the Pre-Block ACL and Post-Block ACL fields.

The Pre-Block ACL is mainly used for permitting what you do not want the sensor to ever block. When a packet is checked against the ACL, the first line that gets matched determines the action. If the first line matched is a permit line from the Pre-Block ACL, the packet is permitted even though there may be a deny line (from an automatic block) listed later in the ACL. The Pre-Block ACL can override the deny lines resulting from the blocks.

The Post-Block ACL is best used for additional blocking or permitting that you want to occur on the same interface or direction. If you have an existing ACL on the interface or direction that the sensor will manage, that existing ACL can be used as a Post-Block ACL. If you do not have a Post-Block ACL, the sensor inserts **permit ip any any** at the end of the new ACL.

When the sensor starts up, it reads the contents of the two ACLs. It creates a third ACL with the following entries:

- A **permit** line for the sensor IP address
- Copies of all configuration lines of the Pre-Block ACL
- A **deny** line for each address being blocked by the sensor
- Copies of all configuration lines of the Post-Block ACL

The sensor applies the new ACL to the interface and direction that you designate.

Note When the new ACL is applied to an interface or direction of the router, it removes the application of any other ACL to that interface or direction.

Configuring the Sensor to Manage Cisco Routers

To configure a sensor to manage a Cisco router to perform blocking and rate limiting, follow these steps:

Step 1 Log in to the CLI using an account with administrator privileges.

Step 2 Enter network access submode:

```
sensor# configure terminal
sensor(config)# service network-access
sensor(config-net)#
```

Step 3 Set the IP address for the router controlled by ARC:

```
sensor(config-net)# router-devices ip_address
```

Step 4 Type the logical device name that you created in Configuring User Profiles, page 10-19.

```
sensor(config-net-rou)# profile-name user_profile_name
```

ARC accepts anything you enter. It does not check to see if the user profile exists.

Step 5 Designate the method used to access the sensor:

```
sensor(config-net-rou)# communication {telnet | ssh-des | ssh-3des}
```

If unspecified, SSH 3DES is used.

Note If you are using DES or 3DES, you must use the command **ssh host-key** *ip_address* to accept the key or ARC cannot connect to the device. For the procedure, see Adding Hosts to the Known Hosts List, page 4-32.

Step 6 Specify the sensor NAT address:

```
sensor(config-net-rou)# nat-address nat_address
```

Note This changes the IP address in the first line of the ACL from the sensor's address to the NAT address. This is not a NAT address configured on the device being managed. It is the address the sensor is translated to by an intermediate device, one that is between the sensor and the device being managed.

Step 7 Specify whether the router will perform blocking, rate limiting, or both:

Note The default is blocking. You do not have to configure response capabilities if you want the router to perform blocking only.

a. For rate limiting only:

```
sensor(config-net-rou)# response-capabilities rate-limit
```

b. For both blocking and rate limiting:

```
sensor(config-net-rou)# response-capabilities block|rate-limit
```

Step 8 Set the interface name and direction:

```
sensor(config-net-rou)# block-interfaces interface_name {in | out}
```

Caution The name of the interface must either be the complete name of the interface or an abbreviation that the router recognizes with the **interface** command.

Step 9 (Optional) Add the pre-ACL name (blocking only):

```
sensor(config-net-rou-blo)# pre-acl-name pre_acl_name
```

Step 10 (Optional) Add the post-ACL name (blocking only):

```
sensor(config-net-rou-blo)# post-acl-name post_acl_name
```

Step 11 Verify the settings:

```
sensor(config-net-rou-blo)# exit
sensor(config-net-rou)# show settings
   ip-address: 10.89.127.97
   ----------------------------------------------
      communication: ssh-3des default: ssh-3des
      nat-address: 19.89.149.219 default: 0.0.0.0
      profile-name: PROFILE1
      block-interfaces (min: 0, max: 100, current: 1)
      ---------------------------------------------
         interface-name: GigabitEthernet0/1
         direction: in
         ---------------------------------------------
            pre-acl-name: <defaulted>
            post-acl-name: <defaulted>
         ---------------------------------------------
      ---------------------------------------------
      response-capabilities: block|rate-limit default: block
   ----------------------------------------------
sensor(config-net-rou)#
```

Step 12 Exit network access submode:

```
sensor(config-net-rou)# exit
sensor(config-net)# exit
sensor(config)# exit
Apply Changes:?[yes]:
```

Step 13 Press **Enter** to apply the changes or enter **no** to discard them.

Configuring the Sensor to Manage Catalyst 6500 Series Switches and Cisco 7600 Series Routers

This section describes how to configure the sensor to manage Cisco switches. It contains the following topics:

Switches and VACLs

You can configure ARC to block using VACLs on the switch itself when running Cisco Catalyst software, or to block using router ACLs on the MSFC or on the switch itself when running Cisco IOS software. This section describes blocking using VACLs. You cannot configure switches that use VACLs to perform rate limiting.

For blocking using the router ACLS, see Configuring the Sensor to Manage Cisco Routers, page 10-22.

You must configure the blocking interfaces on the Catalyst 6500 series switch and specify the VLAN of traffic you want blocked.

You create and save Pre-Block and Post-Block VACLs in your switch configuration. These VACLs must be extended IP VACLs, either named or numbered. See your switch documentation for more information on creating VACLs.

Enter the names of these VACLs that are already configured on your switch in the Pre-Block VACL and Post-Block VACL fields.

The Pre-Block VACL is used mainly for permitting what you do not want the sensor to ever block. When a packet is checked against the VACL, the first line that gets matched determines the action. If the first line matched is a permit line from the Pre-Block VACL, the packet is permitted even though there may be a deny line (from an automatic block) listed later in the VACL. The Pre-Block VACL can override the deny lines resulting from the blocks.

The Post-Block VACL is best used for additional blocking or permitting that you want to occur on the same VLAN. If you have an existing VACL on the VLAN that the sensor will manage, the existing VACL can be used as a Post-Block VACL. If you do not have a Post-Block V ACL, the sensor inserts **permit ip any any** at the end of the new VACL.

Note IDSM-2 inserts **permit ip any any capture** at the end of the new VACL.

When the sensor starts up, it reads the contents of the two VACLs. It creates a third VACL with the following entries:

- A **permit** line for the sensor's IP address
- Copies of all configuration lines of the Pre-Block VACL
- A **deny** line for each address being blocked by the sensor
- Copies of all configuration lines of the Post-Block VACL

The sensor applies the new VACL to the VLAN that you designate.

Note When the new VACL is applied to a VLAN of the switch, it removes the application of any other VACL to that VLAN.

Configuring the Sensor to Manage Catalyst 6500 Series Switches and Cisco 7600 Series Routers

To configure the sensor to manage Catalyst 6500 series switches and Cisco 7600 series routers, follow these steps:

Step 1 Log in to the CLI using an account with administrator privileges.

Step 2 Enter network access submode:

```
sensor# configure terminal
sensor(config)# service network-access
sensor(config-net)#
```

Step 3 Set the IP address for the router controlled by ARC:

```
sensor(config-net)# cat6k-devices ip_address
```

Step 4 Type the user profile name that you created in Configuring User Profiles, page 10-19.

```
sensor(config-net-cat)# profile-name user_profile_name
```

Note ARC accepts anything you enter. It does not check to see if the logical device exists.

Step 5 Designate the method used to access the sensor:

```
sensor(config-net-cat)# communication {telnet | ssh-des/ | sh-3des}
```

If unspecified, SSH 3DES is used.

Note If you are using DES or 3DES, you must use the command **ssh host-key** *ip_address* to accept the key or ARC cannot connect to the device. For the procedure, see Adding Hosts to the Known Hosts List, page 4-32.

Step 6 Specify the sensor NAT address:

```
sensor(config-net-cat)# nat-address nat_address
```

Note This changes the IP address in the first line of the ACL from the sensor's address to the NAT address. This is not a NAT address configured on the device being managed. It is the address the sensor is translated to by an intermediate device, one that is between the sensor and the device being managed.

Step 7 Specify the VLAN number:

```
sensor(config-net-cat)# block-vlans vlan_number
```

Step 8 (Optional) Add the pre-VACL name:

```
sensor(config-net-cat-blo)# pre-vacl-name pre_vacl_name
```

Step 9 (Optional) Add the post-VACL name:

```
sensor(config-net-cat-blo)# post-vacl-name post_vacl_name
```

Step 10 Exit network access submode:

```
sensor(config-net-cat-blo)# exit
sensor(config-net-cat)# exit
sensor(config-net)# exit
sensor(config)# exit
Apply Changes:?[yes]:
```

Step 11 Press **Enter** to apply the changes or enter **no** to discard them.

Configuring the Sensor to Manage Cisco Firewalls

To configure the sensor to manage Cisco firewalls, follow these steps:

Step 1 Log in to the CLI using an account with administrator privileges.

Step 2 Enter network access submode:

```
sensor# configure terminal
sensor(config)# service network-access
sensor(config-net)#
```

Step 3 Set the IP address for the firewall controlled by ARC:

```
sensor(config-net)# firewall-devices ip_address
```

Step 4 Type the user profile name that you created in Configuring User Profiles, page 10-19.

```
sensor(config-net-fir)# profile-name user_profile_name
```

Note ARC accepts anything you enter. It does not check to see if the logical device exists.

Step 5 Designate the method used to access the sensor:

```
sensor(config-net-fir)# communication {telnet | ssh-des | sh-3des}
```

If unspecified, SSH 3DES is used.

Note If you are using DES or 3DES, you must use the command **ssh host-key** *ip_address* to accept the key or ARC cannot connect to the device. For the procedure, see Adding Hosts to the Known Hosts List, page 4-32.

Step 6 Specify the sensor's NAT address:

```
sensor(config-net-fir)# nat-address nat_address
```

Note This changes the IP address in the first line of the ACL from the sensor's address to the NAT address. This is not a NAT address configured on the device being managed. It is the address the sensor is translated to by an intermediate device, one that is between the sensor and the device being managed.

Step 7 Exit network access submode:

```
sensor(config-net-fir)# exit
sensor(config-net)# exit
sensor(config)# exit
Apply Changes:?[yes]:
```

Step 8 Press **Enter** to apply the changes or enter **no** to discard them.

Configuring the Sensor to be a Master Blocking Sensor

Multiple sensors (blocking forwarding sensors) can forward blocking requests to a specified master blocking sensor, which controls one or more devices. The master blocking sensor is the ARC running on a sensor that controls blocking on one or more devices on behalf of one or more other sensors. The ARC on a master blocking sensor controls blocking on devices at the request of the ARCs running on other sensors.

Caution Two sensors cannot control blocking or rate limiting on the same device. If this situation is needed, configure one sensor as the master blocking sensor to manage the devices and the other sensors can forward their requests to the master blocking sensor.

Note When you add a master blocking sensor, you reduce the number of blocking devices per sensor. For example, if you want to block on 10 firewalls and 10 routers with one blocking interface/direction each, you can assign 10 to the sensor and assign the other 10 to a master blocking sensor.

Master blocking sensors can also forward rate limits.

On the blocking forwarding sensor, identify which remote host serves as the master blocking sensor; on the master blocking sensor you must add the blocking forwarding sensors to its access list.

If the master blocking sensor requires TLS for web connections, you must configure the ARC of the blocking forwarding sensor to accept the X.509 certificate of the master blocking sensor remote host. Sensors by default have TLS enabled, but you can change this option.

Note Typically the master blocking sensor is configured to manage the network devices. Blocking forwarding sensors are not normally configured to manage other network devices, although doing so is permissible.

Even if you have no devices configured for blocking or rate limiting, a sensor that is configured for blocking or rate limiting can forward blocking and rate limiting requests to a master blocking sensor. When a signature fires that has blocking or rate limit requests configured as event actions, the sensor forwards the block or rate limit request to the master blocking sensor, which then performs the block or rate limit.

Caution Only one sensor should control all blocking interfaces on a device.

Use the **master-blocking-sensors** *mbs_ip_address* command in the service network access submode to configure a master blocking sensor.

The following options apply:

- *mbs_ip_address*—IP address of sensor for forward block requests.
- **password**—Account password of sensor for forward block requests.
- **port**—Port of sensor for forward block requests.
- **tls {true | false}**—Set to true if the remote sensor requires TLS; otherwise set to false.
- **username**—Account name of sensor for forward block requests.

To configure ARC on a sensor to forward blocks to a master blocking sensor, follow these steps:

Step 1 Log in to the CLI using an account with administrator privileges on both the master blocking sensor and the blocking forwarding sensor.

Step 2 Enter configuration mode on both sensors:

```
sensor# configure terminal
```

Step 3 Configure TLS if necessary:

a. On the master blocking sensor, check to see if it requires TLS and what port number is used:

```
sensor(config)# service web-server
sensor(config-web)# show settings
   enable-tls: true <defaulted>
   port: 443 <defaulted>
   server-id: HTTP/1.1 compliant <defaulted>
sensor(config-web)#
```

If `enable-tls` is true, go to Step b.

b. On the blocking forwarding sensor, configure it to accept the X.509 certificate of the master blocking sensor:

```
sensor(config-web)# exit
sensor(config)# tls trusted-host ip-address mbs_ip_address port port_number
```

Example:

```
sensor(config)# tls trusted-host ip-address 10.0.0.0 port 8080
Certificate MD5 fingerprint is
F4:4A:14:BA:84:F4:51:D0:A4:E2:15:38:7E:77:96:D8Certificate SHA1 fingerprint is
84:09:B6:85:C5:43:60:5B:37:1E:6D:31:6A:30:5F:7E:4D:4D:E8:B2
Would you like to add this to the trusted certificate table for this host?[yes]:
```

Note You are prompted to accept the certificate based on the certificate fingerprint. Sensors provide only self-signed certificates (instead of certificates signed by a recognized certificate authority). You can verify the master blocking sensor host sensor's certificate by logging in to the host sensor and typing the **show tls fingerprint** command to see that the host certificate's fingerprints match.

Step 4 Type **yes** to accept the certificate from the master blocking sensor.

Step 5 Enter network access mode:

```
sensor(config)# service network-access
```

Step 6 Enter general submode:

```
sensor(config-net)# general
```

Step 7 Add a master blocking sensor entry:

```
sensor(config-net-gen)# master-blocking-sensors mbs_ip_address
```

Step 8 Specify the username for an administrative account on the master blocking sensor host:

```
sensor(config-net-gen-mas)# username username
```

Step 9 Specify the password for the user:

```
sensor(config-net-gen-mas)# password
Enter password []: *****
Re-enter mbs-password []: *****
sensor(config-net-gen-mas)#
```

Step 10 Specify the port number for the HTTP communications of the host.

```
sensor(config-net-gen-mas)# port port_number
```

The default is 80/443 if not specified.

Step 11 Set the status of whether or not the host uses TLS/SSL:

```
sensor(config-net-gen-mas)# tls {true | false}
sensor(config-net-gen-mas)
```

Note If you set the value to true, you need to use the command **tls trusted-host ip-address** *mbs_ip_address*.

Step 12 Exit network access submode:

```
sensor(config-net-gen-mas)# exit
sensor(config-net-gen)# exit
sensor(config-net)# exit
sensor(config)# exit
Apply Changes:?[yes]:
```

Step 13 Press **Enter** to apply the changes or enter **no** to discard them.

Step 14 On the master blocking sensor, add the block forwarding sensor's IP address to the access list. For the procedure, see Changing the Access List, page 4-5.

Configuring Manual Blocking

Use the **block-hosts** and **block-networks** commands in the service network access submode to manually block a host or a network. You must have blocking configured before you can set up manual blocks. You can also view a list of hosts and networks that are being blocked.

Note Manual blocks in the CLI are actually changes to the configuration, so they are permanent. You cannot do a timed manual block. You cannot use the IPS manager to delete blocks created by the CLI. Manual blocks have to be removed in the CLI.

Caution We recommend that you use manual blocking on a very limited basis, if at all.

To manually block a host or a network, follow these steps:

Step 1 Log in to the CLI using an account with administrator privileges.

Step 2 Enter network access mode:

```
sensor# configuration terminal
sensor(config)# service network-access
sensor(config-net)#
```

Step 3 Enter general mode:

```
sensor (config-net)# general
sensor(config-net-gen)#
```

Step 4 Start the manual block:

a. For a host IP address:

```
sensor(config-net-gen)# block-hosts ip_address
```

b. For a network IP address:

```
sensor(config-net-gen)# block-networks ip_address/netmask
```

The format for *ip_address/netmask* is A.B.C.D/nn.

Example:

```
sensor (config-net-gen)# block-networks 10.0.0.0/8
```

Note You must end the manual block in the CLI or it is permanent.

Step 5 To end the manual block:

```
sensor (config-net-gen)# no block-hosts ip_address
```

Step 6 Exit network access submode:

```
sensor (config-net-gen)# exit
sensor (config-net)# exit
sensor(config)# exit
sensor#
```

Obtaining a List of Blocked Hosts and Connections

Use the **show statistics** command to obtain a list of blocked hosts and blocked connections.

To obtain a list of blocked hosts and connections, follow these steps:

Step 1 Log in to the CLI.

Step 2 Check the statistics for ARC:

```
sensor# show statistics network-access
Current Configuration
   LogAllBlockEventsAndSensors = true
   EnableNvramWrite = false
   EnableAclLogging = false
   AllowSensorBlock = false
   BlockMaxEntries = 250
   MaxDeviceInterfaces = 250
   NetDevice
      Type = Cisco
      IP = 10.1.1.1
      NATAddr = 0.0.0.0
      Communications = telnet
      BlockInterface
         InterfaceName = fa0/0
         InterfaceDirection = in
State
   BlockEnable = true
   NetDevice
      IP = 10.1.1.1
      AclSupport = uses Named ACLs
      Version = 12.2
      State = Active
   BlockedAddr
      Host
         IP = 192.168.1.1
         Vlan =
         ActualIp =
         BlockMinutes = 80
         MinutesRemaining = 76
```

The `Host` entry indicates which hosts are being blocked and how long the blocks are.

CHAPTER 11

Configuring SNMP

Note To have the sensor send SNMP traps, you must also select **request-snmp-trap** as the event action when you configure signatures. For more information, see Assigning Actions to Signatures, page 7-11.

This chapter describes how to configure SNMP. It contains the following sections:

Understanding SNMP

SNMP is an application layer protocol that facilitates the exchange of management information between network devices. SNMP enables network administrators to manage network performance, find and solve network problems, and plan for network growth.

SNMP is a simple request/response protocol. The network-management system issues a request, and managed devices return responses. This behavior is implemented by using one of four protocol operations: Get, GetNext, Set, and Trap.

You can configure the sensor for monitoring by SNMP. SNMP defines a standard way for network management stations to monitor the health and status of many types of devices, including switches, routers, and sensors.

You can configure the sensor to send SNMP traps. SNMP traps enable an agent to notify the management station of significant events by way of an unsolicited SNMP message.

Trap-directed notification has the following advantage—if a manager is responsible for a large number of devices, and each device has a large number of objects, it is impractical to poll or request information from every object on every device. The solution is for each agent on the managed device to notify the manager without solicitation. It does this by sending a message known as a trap of the event.

After receiving the event, the manager displays it and can take an action based on the event. For instance, the manager can poll the agent directly, or poll other associated device agents to get a better understanding of the event.

Note Trap-directed notification results in substantial savings of network and agent resources by eliminating frivolous SNMP requests. However, it is not possible to totally eliminate SNMP polling. SNMP requests are required for discovery and topology changes. In addition, a managed device agent cannot send a trap if the device has had a catastrophic outage.

Configuring SNMP

Note To have the sensor send SNMP traps, you must also select **request-snmp-trap** as the event action when you configure signatures. See For more information, see Assigning Actions to Signatures, page 7-11.

Configure general SNMP parameters in the service notification submode.

The following options apply:

- **default**—Sets the value back to the system default setting.
- **enable-set-get {true | false}**—Enables the gets and sets of object identifiers (OIDs).
- **no**—Remove an entry or selection setting.
- **read-only-community**—The read-only community name for the SNMP agent.

  The default is public.
- **read-write-community**—The read-write community name for the SNMP agent.

  The default is private.
- **snmp-agent-port**—The port the SNMP agent will listen on.

  The default SNMP port number is 161.
- **snmp-agent-protocol**—The protocol the SNMP agent will communicate with.

  The default protocol is UDP.
- **system-contact**—The contact information for this sensor.

  The system-contact option modifies the SNMPv2-MIB::sysContact.0 value.
- **system-location**—The location of the sensor.

  The system-location option modifies the SNMPv2-MIB::sysLocation.0 value.

To configure SNMP general parameters, follow these steps:

Step 1 Log in to the CLI using an account with administrator privileges.

Step 2 Enter notification submode:

```
sensor# configure terminal
sensor(config)# service notification
sensor(config-not)#
```

Step 3 Enable SNMP so that the SNMP management workstation can issue requests to the sensor SNMP agent:

```
sensor(config-not)# enable-set-get true
```

Step 4 Configure the SNMP agent parameters:

These values configure the community name on the sensor SNMP agent. A community name is a plain-text password mechanism that is used to weakly authenticate SNMP queries.

a. Assign the read-only community string:

```
sensor(config-not)# read-only-community PUBLIC1
```

The read-only community name specifies the password for queries to the SNMP agent.

b. Assign the read-write community string:

```
sensor(config-not)# read-write-community PRIVATE1
```

The read-write community name specifies the password for sets to the SNMP agent.

Note The management workstation sends SNMP requests to the sensor SNMP agent, which resides on the sensor. If the management workstation issues a request and the community string does not match what is on the senor, the sensor rejects it.

c. Assign the sensor contact user ID:

```
sensor(config-not)# system-contact BUSINESS
```

d. Type the location of the sensor:

```
sensor(config-not)# system-location AUSTIN
```

e. Type the port of the sensor SNMP agent:

```
sensor(config-not)# snmp-agent-port 161
```

Note You must reboot the sensor if you change the port or protocol.

f. Select the protocol the sensor SNMP agent will use:

```
sensor(config-not)# snmp-agent-protocol udp
```

Note You must reboot the sensor if you change the port or protocol.

Step 5 Verify the settings:

```
sensor(config-not)# show settings
   trap-destinations (min: 0, max: 10, current: 0)
   -----------------------------------------------
   -----------------------------------------------
   error-filter: error|fatal <defaulted>
   enable-detail-traps: false <defaulted>
   enable-notifications: false <defaulted>
   enable-set-get: true default: false
   snmp-agent-port: 161 default: 161
   snmp-agent-protocol: udp default: udp
   read-only-community: PUBLIC1 default: public
   read-write-community: PRIVATE1 default: private
   trap-community-name: public <defaulted>
   system-location: AUSTIN default: Unknown
   system-contact: BUSINESS default: Unknown
sensor(config-not)#
```

Step 6 Exit notification submode:

```
sensor(config-not)# exit
Apply Changes:?[yes]:
```

Step 7 Press **Enter** to apply the changes or enter **no** to discard them.

Configuring SNMP Traps

Note To have the sensor send SNMP traps, you must also select **request-snmp-trap** as the event action when you configure signatures. For more information, see Assigning Actions to Signatures, page 7-11.

Configure the SNMP traps in the service notification submode.

The following options apply:

- **enable-detail-traps {true | false}**—Enables the sending of detailed traps with no size limit. Otherwise traps are sent in sparse mode (less than 484 bytes).
- **enable-notifications {true | false}**—Enables event notifications.
- **error-filter {warning | error | fatal}**—Determines which errors generate an SNMP trap. An SNMP trap is generated for every evError event that matches the filter. The default is error and fatal.
- **trap-community-name**—The community name used when sending traps if no name is specified when defining the trap destinations.
- **trap-destinations**—Defines the destinations to send error events and alert events generated from signature actions.
  - **trap-community-name**—The community name used when sending the trap. If no community name is specified the general trap community name is used.
  - **trap-port**—The port number to send the SNMP trap to.

To configure SNMP traps, follow these steps:

Step 1 Log in to the CLI using an account with administrator privileges.

Step 2 Enter notification submode:

```
sensor# configure terminal
sensor(config)# service notification
sensor(config-not)#
```

Step 3 Enable SNMP traps:

```
sensor(config-not)# enable-notifications true
```

Step 4 Set the parameters for the SNMP trap:

a. Select the error events you want to be notified about through SNMP traps:

```
sensor(config-not)# error-filter {error | warning | fatal}
```

Note The **error-filter {error | warning | fatal}** command includes error, warning, and fatal traps. It filters in (not filters out) the traps based on severity.

b. Choose whether you want detailed SNMP traps:

```
sensor(config-not)# enable-detail-traps true
```

c. Type the community string to be included in the detailed traps:

```
sensor(config-not)# trap-community-name TRAP1
```

Step 5 Set the parameters for the SNMP trap destinations so the sensor knows which management workstations to send them to:

a. Type the IP address of the SNMP management station:

```
sensor(config-not)# trap-destinations 10.1.1.1
```

b. Type the UDP port of the SNMP management station:

```
sensor(config-not-tra)# trap-port 162
```

The default is 162.

c. Type the trap Community string:

```
sensor(config-not-tra)# trap-community-name AUSTIN_PUBLI
```

Note The community string appears in the trap and is useful if you are receiving multiple types of traps from multiple agents. For example, a router or sensor could be sending the traps, and if you put something that identifies the router or sensor specifically in your community string, you can filter the traps based on the community string.

Step 6 Verify the settings:

```
sensor(config-not-tra)# exit
sensor(config-not)# show settings
   trap-destinations (min: 0, max: 10, current: 1)
   -----------------------------------------------
      ip-address: 10.1.1.1
      ---------------------------------------------
         trap-community-name: AUSTIN_PUBLIC default:
         trap-port: 161 default: 162
      ---------------------------------------------
   -----------------------------------------------
   error-filter: warning|error|fatal default: error|fatal
   enable-detail-traps: true default: false
   enable-notifications: true default: false
   enable-set-get: true default: false
   snmp-agent-port: 161 default: 161
   snmp-agent-protocol: udp default: udp
   read-only-community: PUBLIC1 default: public
   read-write-community: PRIVATE1 default: private
   trap-community-name: PUBLIC1 default: public
   system-location: AUSTIN default: Unknown
   system-contact: BUSINESS default: Unknown
sensor(config-not)#
```

Step 7 Exit notification submode:

```
sensor(config-not)# exit
```

```
Apply Changes:?[yes]:
```

Step 8 Press **Enter** to apply the changes or enter **no** to discard them.

Supported MIBS

The following private MIBS are supported on the sensor:

- CISCO-CIDS-MIB
- CISCO-ENHANCED-MEMPOOL-MIB
- CISCO-ENTITY-ALARM-MIB

You can obtain these private Cisco MIBs under the heading SNMP v2 MIBs at this URL:

http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

Note MIB II is available on the sensor, but we do not support it. We know that some elements are not correct (for example, the packet counts from the IF MIB on the sensing interfaces). While you can use elements from MIB II, we do not guarantee that they all provide correct information. We fully support the other listed MIBs and their output is correct.

Note CISCO-PROCESS-MIB is available on the sensor, but we do not support it. We know that some elements are not available. While you can use elements from CISCO-PROCESS-MIB, we do not guarantee that they all provide correct information. We fully support the other listed MIBs and their output is correct.

CHAPTER 12

Working With Configuration Files

This chapter describes how to use commands that show, copy, and erase the configuration file. It contains the following sections:

Displaying the Current Configuration

Use the **show configuration** or the **more current-config** command to display the contents of the current configuration.

To display the contents of the current configuration, follow these steps:

Step 1 Log in to the CLI.

Step 2 Display the current configuration:

```
sensor# show configuration
! ------------------------------
! Version 5.1(0.7)
! Current configuration last modified Thu Jul 14 21:49:58 2005
! ------------------------------
display-serial
! ------------------------------
service interface
exit
! ------------------------------
service analysis-engine
exit
! ------------------------------
service authentication
exit
! ------------------------------
```

```
service event-action-rules rules0
exit
! ------------------------------
service host
network-settings
host-ip 10.89.149.27/25,10.89.149.126
host-name sensor
telnet-option enabled
access-list 10.0.0.0/8
access-list 64.0.0.0/8
exit
time-zone-settings
offset 0
standard-time-zone-name UTC
exit
exit
! ------------------------------
service logger
exit
! ------------------------------
service network-access
user-profiles test
exit
exit
! ------------------------------
service notification
exit
! ------------------------------
service signature-definition sig0
signatures 60000 0
alert-severity medium
sig-fidelity-rating 75
sig-description
sig-name My Sig
sig-string-info My Sig Info
sig-comment Sig Comment
exit
engine string-tcp
event-action produce-alert
direction to-service
regex-string My Regex String
service-ports 23
exit
event-counter
event-count 1
event-count-key Axxx
specify-alert-interval no
exit
alert-frequency
summary-mode summarize
summary-interval 15
summary-key Axxx
specify-global-summary-threshold yes
global-summary-threshold 75
exit
exit
exit
exit
exit
! ------------------------------
service ssh-known-hosts
exit
! ------------------------------
service trusted-certificates
```

```
exit
! ------------------------------
service web-server
exit
sensor#
```

Displaying the Current Submode Configuration

Use the **show settings** command in a submode to display the current configuration of that submode.

To display the current configuration of a submode, follow these steps:

Step 1 Log in to the CLI.

Step 2 Display the current configuration of the service analysis engine submode:

```
sensor# configure terminal
sensor(config)# service analysis-engine
sensor(config-ana)# show settings
   global-parameters
   ----------------------------------------------
      ip-logging
      ----------------------------------------------
         max-open-iplog-files: 20 <defaulted>
      ----------------------------------------------
   ----------------------------------------------
   virtual-sensor (min: 1, max: 255, current: 1)
   ----------------------------------------------
      <protected entry>
      name: vs0 <defaulted>
      ----------------------------------------------
         description: default virtual sensor <defaulted>
         signature-definition: sig0 <protected>
         event-action-rules: rules0 <protected>
         physical-interface (min: 0, max: 999999999, current: 0)
         ----------------------------------------------
         ----------------------------------------------
         logical-interface (min: 0, max: 999999999, current: 0)
         ----------------------------------------------
         ----------------------------------------------
      ----------------------------------------------
   ----------------------------------------------
sensor(config-ana)# exit
sensor(config)# exit
sensor#
```

Step 3 Display the current configuration of the service authentication submode:

```
sensor# configure terminal
sensor(config)# service authentication
sensor(config-aut)# show settings
   attemptLimit: 0 <defaulted>
sensor(config-aut)# exit
sensor(config)# exit
sensor#
```

Step 4 Display the current configuration of the service event-action-rules submode:

```
sensor# configure terminal
```

```
sensor(config)# service event-action-rules rules0
sensor(config-rul)# show settings
   variables (min: 0, max: 256, current: 0)
   ----------------------------------------------
   ----------------------------------------------
   overrides (min: 0, max: 12, current: 0)
   ----------------------------------------------
   ----------------------------------------------
   filters (min: 0, max: 4096, current: 0 - 0 active, 0 inactive)
   ----------------------------------------------
   general
   ----------------------------------------------
      global-overrides-status: Enabled <defaulted>
      global-filters-status: Enabled <defaulted>
      global-summarization-status: Enabled <defaulted>
      global-metaevent-status: Enabled <defaulted>
      global-deny-timeout: 3600 <defaulted>
      global-block-timeout: 30 <defaulted>
      max-denied-attackers: 10000 <defaulted>
   ----------------------------------------------
   target-value (min: 0, max: 5, current: 0)
   ----------------------------------------------
   ----------------------------------------------
sensor(config-rul)# exit
sensor(config)# exit
sensor# exit
```

Step 5 Display the current configuration of the service host submode:

```
sensor# configure terminal
sensor(config)# service host
sensor(config-hos)# show settings
   network-settings
   ----------------------------------------------
      host-ip: 10.89.149.27/25,10.89.149.126 default: 10.1.9.201/24,10.1.9.1
      host-name: sensor default: sensor
      telnet-option: enabled default: disabled
      access-list (min: 0, max: 512, current: 2)
      ---------------------------------------------
         network-address: 10.0.0.0/8
         ---------------------------------------------
         network-address: 64.0.0.0/8
         ---------------------------------------------
      ---------------------------------------------
      ftp-timeout: 300 seconds <defaulted>
      login-banner-text: <defaulted>
   ----------------------------------------------
   time-zone-settings
   ----------------------------------------------
      offset: 0 minutes default: 0
      standard-time-zone-name: UTC default: UTC
   ----------------------------------------------
   ntp-option
   ----------------------------------------------
      disabled
      ---------------------------------------------
      ---------------------------------------------
   ----------------------------------------------
   summertime-option
   ----------------------------------------------
      disabled
      ---------------------------------------------
      ---------------------------------------------
   ----------------------------------------------
```

```
   auto-upgrade-option
   -----------------------------------------------
      disabled
      ----------------------------------------------
      ----------------------------------------------
   -----------------------------------------------
   crypto
   -----------------------------------------------
      key (min: 0, max: 10, current: 2)
      ----------------------------------------------
         <protected entry>
         name: realm-cisco.pub <defaulted>
         type
         ----------------------------------------------
            rsa-pubkey
            ----------------------------------------------
               length: 2048 <defaulted>
               exponent: 65537 <defaulted>
               modulus: 24442189989357747083874855335232628843599968934198559648
630199473878411519325039111726689401947545491553904076580203933306118912925083 00
859403040311860144996325688124280680580895816141963373996230606249900570491030 55
901539559350860600086797768080736401860634357232523755752931263045580687043018 63
805621144374392890694566709220749958273902847616105915157520084051402436730831 89
778224699649345983670103893898882974908028841185437300762935897035359121619933 19
470931302986888300125472155726463496235394688386410649153139478068529040823519 55
132172731380999653830397161301532707152200465671078281289241976924173320339117 04
3 <defaulted>
            ----------------------------------------------
         ----------------------------------------------
         <protected entry>
         name: realm-trend.pub <defaulted>
         type
         ----------------------------------------------
            rsa-pubkey
            ----------------------------------------------
               length: 2048 <defaulted>
               exponent: 65537 <defaulted>
               modulus: 21765561422573021314159855351418723031625093380777053696
638172895270605709325510654898181907137456721482605270300606672083666066038026 79
304390667241433906264954793005501016181795846372870529364656921465726126513759 69
203545215856442216029442035208044042129754019708951199037567696011338536732967 66
452897957779734919840565870452145148200633669507313464000443084915946264347069 99
476086688228140148300633995342046470695090524434395253637065272552245107711222 35
801811504605447832514984814327059910100698443685257548784136694276397529508017 67
999053092352324562955800867242032979140959842243284443915822231384237991008381 91
9 <defaulted>
            ----------------------------------------------
         ----------------------------------------------
      ----------------------------------------------
   ----------------------------------------------
sensor(config-hos)# exit
sensor(config)# exit
sensor#
```

Step 6 Display the current configuration of the service interface submode:

```
sensor# configure terminal
sensor(config)# service interface
sensor(config-int)# show settings
   physical-interfaces (min: 0, max: 999999999, current: 4)
   ----------------------------------------------
      <protected entry>
      name: GigabitEthernet0/0 <defaulted>
      ----------------------------------------------
```

```
   media-type: tx <protected>
   description: <defaulted>
   admin-state: disabled <defaulted>
   duplex: auto <defaulted>
   speed: auto <defaulted>
   alt-tcp-reset-interface
   ----------------------------------------------
      none
      ---------------------------------------------
      ---------------------------------------------
   ----------------------------------------------
   subinterface-type
   ----------------------------------------------
      none
      ---------------------------------------------
      ---------------------------------------------
   ----------------------------------------------
----------------------------------------------
<protected entry>
name: GigabitEthernet0/1 <defaulted>
----------------------------------------------
   media-type: tx <protected>
   description: <defaulted>
   admin-state: disabled <protected>
   duplex: auto <defaulted>
   speed: auto <defaulted>
   alt-tcp-reset-interface
   ----------------------------------------------
      none
      ---------------------------------------------
      ---------------------------------------------
   ----------------------------------------------
   subinterface-type
   ----------------------------------------------
      none
      ---------------------------------------------
      ---------------------------------------------
   ----------------------------------------------
----------------------------------------------
<protected entry>
name: GigabitEthernet2/0 <defaulted>
----------------------------------------------
   media-type: xl <protected>
   description: <defaulted>
   admin-state: disabled <defaulted>
   duplex: auto <defaulted>
   speed: auto <defaulted>
   alt-tcp-reset-interface
   ----------------------------------------------
      none
      ---------------------------------------------
      ---------------------------------------------
   ----------------------------------------------
   subinterface-type
   ----------------------------------------------
      none
      ---------------------------------------------
      ---------------------------------------------
   ----------------------------------------------
----------------------------------------------
<protected entry>
name: GigabitEthernet2/1 <defaulted>
----------------------------------------------
   media-type: xl <protected>
```

```
            description: <defaulted>
            admin-state: disabled <defaulted>
            duplex: auto <defaulted>
            speed: auto <defaulted>
            alt-tcp-reset-interface
            ------------------------------------------------
               none
               ------------------------------------------------
               ------------------------------------------------
            ------------------------------------------------
            subinterface-type
            ------------------------------------------------
               none
               ------------------------------------------------
               ------------------------------------------------
            ------------------------------------------------
         ------------------------------------------------
      ------------------------------------------------
      command-control: GigabitEthernet0/1 <protected>
      inline-interfaces (min: 0, max: 999999999, current: 0)
      ------------------------------------------------
      ------------------------------------------------
      bypass-mode: auto <defaulted>
      interface-notifications
      ------------------------------------------------
         missed-percentage-threshold: 0 percent <defaulted>
         notification-interval: 30 seconds <defaulted>
         idle-interface-delay: 30 seconds <defaulted>
      ------------------------------------------------
sensor(config-int)# exit
sensor(config)# exit
sensor#
```

Step 7 Display the current configuration for the service logger submode:

```
sensor# configure terminal
sensor(config)# service logger
sensor(config-log)# show settings
   master-control
   ------------------------------------------------
      enable-debug: false <defaulted>
      individual-zone-control: false <defaulted>
   ------------------------------------------------
   zone-control (min: 0, max: 999999999, current: 14)
   ------------------------------------------------
      <protected entry>
      zone-name: Cid
      severity: debug <defaulted>
      <protected entry>
      zone-name: AuthenticationApp
      severity: warning <defaulted>
      <protected entry>
      zone-name: Cli
      severity: warning <defaulted>
      <protected entry>
      zone-name: csi
      severity: warning <defaulted>
      <protected entry>
      zone-name: ctlTransSource
      severity: warning <defaulted>
      <protected entry>
      zone-name: IdapiCtlTrans
      severity: warning <defaulted>
      <protected entry>
```

```
      zone-name: IdsEventStore
      severity: warning <defaulted>
      <protected entry>
      zone-name: MpInstaller
      severity: warning <defaulted>
      <protected entry>
      zone-name: nac
      severity: warning <defaulted>
      <protected entry>
      zone-name: sensorApp
      severity: warning <defaulted>
      <protected entry>
      zone-name: tls
      severity: warning <defaulted>
      <protected entry>
      zone-name: intfc
      severity: warning <defaulted>
      <protected entry>
      zone-name: cmgr
      severity: warning <defaulted>
      <protected entry>
      zone-name: cplane
      severity: warning <defaulted>
   ----------------------------------------------
sensor(config-log)# exit
sensor(config)# exit
sensor#
```

Step 8 Display the current configuration for the service network access submode:

```
sensor# configure terminal
sensor(config)# service network-access
sensor(config-net)# show settings
   general
   ----------------------------------------------
      log-all-block-events-and-errors: true <defaulted>
      enable-nvram-write: false <defaulted>
      enable-acl-logging: false <defaulted>
      allow-sensor-block: false <defaulted>
      block-enable: true <defaulted>
      block-max-entries: 250 <defaulted>
      max-interfaces: 250 <defaulted>
      rate-limit-max-entries: 250 <defaulted>
      master-blocking-sensors (min: 0, max: 100, current: 0)
      ----------------------------------------------
      ----------------------------------------------
      never-block-hosts (min: 0, max: 250, current: 0)
      ----------------------------------------------
      ----------------------------------------------
      never-block-networks (min: 0, max: 250, current: 0)
      ----------------------------------------------
      ----------------------------------------------
      block-hosts (min: 0, max: 250, current: 0)
      ----------------------------------------------
      ----------------------------------------------
      block-networks (min: 0, max: 250, current: 0)
      ----------------------------------------------
      ----------------------------------------------
   ----------------------------------------------
   user-profiles (min: 0, max: 250, current: 1)
   ----------------------------------------------
      profile-name: test
      ----------------------------------------------
         enable-password: <hidden>
```

```
            password: <hidden>
            username: <defaulted>
         ----------------------------------------------
      ---------------------------------------------
      cat6k-devices (min: 0, max: 250, current: 0)
      ---------------------------------------------
      ---------------------------------------------
      router-devices (min: 0, max: 250, current: 0)
      ---------------------------------------------
      ---------------------------------------------
      firewall-devices (min: 0, max: 250, current: 0)
      ---------------------------------------------
      ---------------------------------------------
sensor(config-net)# exit
sensor(config)# exit
sensor#
```

Step 9 Display the current configuration for the notification submode:

```
sensor# configure terminal
sensor(config)# service notification
sensor(config-not)# show settings
   trap-destinations (min: 0, max: 10, current: 0)
   ---------------------------------------------
   ---------------------------------------------
   error-filter: error|fatal <defaulted>
   enable-detail-traps: false <defaulted>
   enable-notifications: false <defaulted>
   enable-set-get: false <defaulted>
   snmp-agent-port: 161 <defaulted>
   snmp-agent-protocol: udp <defaulted>
   read-only-community: public <defaulted>
   read-write-community: private <defaulted>
   trap-community-name: public <defaulted>
   system-location: Unknown <defaulted>
   system-contact: Unknown <defaulted>
sensor(config-not)# exit
sensor(config)# exit
sensor#
```

Step 10 Display the current configuration for the signature definitions submode:

```
sensor# configure terminal
sensor(config)# service signature-definition sig0
sensor(config-sig)# show settings
   variables (min: 0, max: 256, current: 1)
   ---------------------------------------------
      <protected entry>
      variable-name: WEBPORTS
      ----------------------------------------------
         web-ports: 80-80,3128-3128,8000-8000,8010-8010,8080-8080,8888-8888,2432
6-24326 <defaulted>
      ----------------------------------------------
   ---------------------------------------------
   application-policy
   ---------------------------------------------
      http-policy
      ----------------------------------------------
         http-enable: false <defaulted>
         max-outstanding-http-requests-per-connection: 10 <defaulted>
         aic-web-ports: 80-80,3128-3128,8000-8000,8010-8010,8080-8080,8888-8888,
24326-24326 <defaulted>
      ----------------------------------------------
      ftp-enable: false <defaulted>
```

```
   ---------------------------------------------
   fragment-reassembly
   ---------------------------------------------
      ip-reassemble-mode: nt <defaulted>
   ---------------------------------------------
   stream-reassembly
   ---------------------------------------------
--MORE--
```

Step 11 Display the current configuration for the SSH known hosts submode:

```
sensor# configure terminal
sensor(config)# service ssh-known-hosts
sensor(config-ssh)# show settings
   rsa1-keys (min: 0, max: 500, current: 0)
   ---------------------------------------------
   ---------------------------------------------
sensor(config-ssh)# exit
sensor(config)# exit
sensor#
```

Step 12 Display the current configuration for the trusted certificates submode:

```
sensor# configure terminal
sensor(config)# service trusted-certificate
sensor(config-tru)# show settings
   trusted-certificates (min: 0, max: 500, current: 1)
   ---------------------------------------------
      common-name: 10.89.130.108
      certificate: MIICJDCCAY0CCPbSkgXUchJIMA0GCSqGSIb3DQEBBQUAMFcxCzAJBgNVBAYTA
lVTMRwwGgYDVQQKExNDaXNjbyBTeXN0ZW1zLCBJbmMuMRIwEAYDVQQLEwlTU00tSVBTMjAxFjAUBgNVB
AMTDTEwLjg5LjEzMC4xMDgwHhcNMDMwMTAzMDE1MjEwWhcNMDUwMTAzMDE1MjEwWjBXMQswCQYDVQQGE
wJVUzEcMBoGA1UEChMTQ2lzY28gU3lzdGVtcywgSW5jLjESMBAGA1UECxMJU1NNLUlQUzIwMRYwFAYDV
QQDEw0xMC44OS4xMzAuMTA4MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQCzldqLFG4MT4bfgh3mJ
fP/DCilnnaLfzHK9FdnhmWI4FY+9MVvAI7MOhAcuV6HYfyp6n6cYvH+Eswzl9uv7H5nouID9St9GI3Yr
SUtlIQAJ4QVL2DwWP230x6KdHrYqcj+Nmhc7AnnPypjidwGSfF+VetIJLEeRFh/mI2JcmwF2QIDAQABM
A0GCSqGSIb3DQEBBQUAA4GBAAUI2PLANTOehxvCfwd6UAFXvy8uifbjqKMC1jrrF+f9KGkxmR+XZvUaG
OS83FYDXlXJvB5Xyxms+Y01wGjzKKpxegBoan8OB8o193Ueszdpvz2xYmiEgywCDyVJRsw3hAFMXWMS5
XsBUiHtw0btHH0j7ElFZxUjZv12fGz8hlnY
   ---------------------------------------------
sensor(config-tru)# exit
sensor(config)# exit
sensor#
```

Step 13 Display the current configuration for the web server submode:

```
sensor# configure terminal
sensor(config)# service web-server
sensor(config-web)# show settings
   enable-tls: true <defaulted>
   port: 443 <defaulted>
   server-id: HTTP/1.1 compliant <defaulted>
sensor(config-web)# exit
sensor(config)# exit
sensor#
```

Filtering the Current Configuration Output

Use the **show configuration | [begin | exclude | include]** *regular_expression* command to search or filter the output of the contents of the current configuration.

Note Users with operator or viewer privileges can search or filter the **current-config** only.

The following options apply:

- **|**—The pipe symbol indicates that an output processing specification follows.
- **begin**—Begins unfiltered output of the **show configuration** command with the first line that contains the regular expression specified.
- **exclude**—Excludes lines in the output of the **show configuration** command that contain a particular regular expression.
- **include**—Includes only the lines in the output of the **show configuration** command that contain the regular expression you specify.
- *regular_expression*—Any regular expression found in the **show configuration** command output.

  Note The *regular_expression* option is case sensitive and allows for complex matching requirements.

To search or filter the output of the contents of the current configuration, follow these steps:

Step 1 Log in to the CLI using an account with administrator privileges.

Step 2 Search the configuration output beginning with the regular expression “ssh,” for example

Note The **show configuration | begin** *regular_expression* command begins unfiltered output of the **show** command with the first line that contains the specified regular expression.

```
sensor# show configuration | begin ssh
communication ssh-3des
profile-name test1
block-vlans 234
pre-vacl-name aaaa
post-vacl-name bbbb
exit
exit
exit
! ------------------------------
service notification
exit
! ------------------------------
service signature-definition sig0
signatures 2200 0
engine service-generic
specify-payload-source yes
payload-source l2-header
exit
exit
exit
signatures 12300 0
```

```
status
enabled true
retired true
--MORE--
```

Note Press **Ctrl-C** to stop the output and return to the CLI prompt.

Step 3 Filter the current configuration so that you exclude lines that contain a regular expression, for example, "service":

```
sensor# show configuration | exclude service
! ------------------------------
! Version 5.1(0.7)
! Current configuration last modified Thu Jul 14 21:49:58 2005
! ------------------------------
display-serial
! ------------------------------
exit
! ------------------------------
exit
! ------------------------------
exit
! ------------------------------
exit
! ------------------------------
network-settings
host-ip 10.89.149.27/25,10.89.149.126
host-name sensor
telnet-option enabled
access-list 10.0.0.0/8
access-list 64.0.0.0/8
exit
time-zone-settings
offset 0
standard-time-zone-name UTC!
--MORE--
```

Note Press **Ctrl-C** to stop the output and return to the CLI prompt.

Step 4 Filter the current configuration so that you include lines that contain a regular expression, for example, "service":

```
sensor# show configuration | include service
service analysis-engine
service authentication
service event-action-rules rules0
service host
service interface
service logger
service network-access
service notification
service signature-definition sig0
```

```
engine service-generic
service ssh-known-hosts
service trusted-certificates
service web-server
sensor#
```

Filtering the Current Submode Configuration Output

Use the **show settings | [begin | exclude | include]** *regular_expression* command in the submode you are interested in to search or filter the output of the contents of the submode configuration.

The following options apply:

- **|**—The pipe symbol indicates that an output processing specification follows.
- **begin**—Begins unfiltered output of the **show settings** command with the first line that contains the regular expression specified.
- **exclude**—Excludes lines in the output of the **show settings** command that contain a particular regular expression.
- **include**—Includes only the lines in the output of the **show settings** command that contain the regular expression you specify.
- *regular_expression*—Any regular expression found in the **show settings** command output.

Note The *regular-expression* option is case sensitive and allows for complex matching requirements.

To search or filter the output of the contents of the submode configuration, follow these steps:

Step 1 Log in to the CLI using an account with administrator privileges.

Step 2 Search the output of the event action rules settings for the regular expression, “filters,” for example:

```
sensor# configure terminal
sensor(config)# service event-action-rules
sensor(config-rul)# show settings | begin filters
filters (min: 0, max: 4096, current: 0 - 0 active, 0 inactive)
 ----------------------------------------------
 general
 ----------------------------------------------
    global-overrides-status: Enabled <defaulted>
    global-filters-status: Enabled <defaulted>
    global-summarization-status: Enabled <defaulted>
    global-metaevent-status: Enabled <defaulted>
    global-deny-timeout: 3600 <defaulted>
    global-block-timeout: 15 default: 30
    max-denied-attackers: 10000 <defaulted>
 ----------------------------------------------
 target-value (min: 0, max: 5, current: 0)
 ----------------------------------------------
 ----------------------------------------------
sensor(config-rul)#
```

Step 3 Filter the output of the network access settings to exclude the regular expression:

```
sensor# configure terminal
sensor(config)# service network-access
sensor(config-net)# show settings | exclude false
   general
   ----------------------------------------------
      log-all-block-events-and-errors: true default: true
      block-enable: true default: true
      block-max-entries: 11 default: 250
      max-interfaces: 13 default: 250
      master-blocking-sensors (min: 0, max: 100, current: 1)
      ----------------------------------------------
         ipaddress: 10.89.149.124
         ----------------------------------------------
            password: <hidden>
            port: 443 default: 443
            tls: true default: true
            username: cisco default:
         ----------------------------------------------
      ----------------------------------------------
      never-block-hosts (min: 0, max: 250, current: 1)
      ----------------------------------------------
         ip-address: 10.89.146.112
         ----------------------------------------------
      ----------------------------------------------
      never-block-networks (min: 0, max: 250, current: 1)
      ----------------------------------------------
         ip-address: 88.88.88.0/24
--MORE--
```

Step 4 Filter the output of the host settings to include the regular expression “ip”:

```
sensor# configure terminal
sensor(config)# service host
sensor(config-hos)# show settings | include ip
      host-ip: 10.89.149.185/25,10.89.149.254 default: 10.1.9.201/24,10.1.9.1
sensor(config-hos)#
```

Displaying the Contents of a Logical File

Use the **more** *keyword* command to display the contents of a logical file, such as the current system configuration or the saved backup system configuration.

The following options apply:

- *keyword*—Either the current-config or the backup-config.
  - **current-config**—The current running configuration. This configuration becomes persistent as the commands are entered.
  - **backup-config**—The storage location for the configuration backup file.

Note Operators and viewers can only display the current configuration. Only administrators can view hidden fields such as passwords.

You can disable the more prompt in **more current-config** or **more backup-config** by setting the terminal length to zero using the **terminal length 0** command. The **more** command then displays the entire file content without pausing. For the procedure for using the **terminal** command, see Modifying Terminal Properties, page 13-3.

To display the contents of a logical file, follow these steps:

Step 1 Log in to the CLI using an account with administrator privileges.

Step 2 Display the contents of the current configuration file:

```
sensor# more current-config
Generating current config:
```

The current configuration is displayed.

```
! ------------------------------
! Version 5.1(0.7)
! Current configuration last modified Thu Jul 14 21:49:58 2005
! ------------------------------
display-serial
! ------------------------------
service interface
exit
! ------------------------------
service analysis-engine
exit
! ------------------------------
service authentication
exit
! ------------------------------
service event-action-rules rules0
exit
! ------------------------------
service host
network-settings
host-ip 10.89.149.27/25,10.89.149.126
host-name sensor
telnet-option enabled
access-list 10.0.0.0/8
access-list 64.0.0.0/8
exit
time-zone-settings
offset 0
standard-time-zone-name UTC
exit
exit
! ------------------------------
service logger
exit
! ------------------------------
service network-access
user-profiles test
exit
exit
! ------------------------------
service notification
exit
! ------------------------------
service signature-definition sig0
signatures 60000 0
alert-severity medium
sig-fidelity-rating 75
```

```
sig-description
sig-name My Sig
sig-string-info My Sig Info
sig-comment Sig Comment
exit
engine string-tcp
event-action produce-alert
direction to-service
regex-string My Regex String
service-ports 23
exit
event-counter
event-count 1
event-count-key Axxx
specify-alert-interval no
exit
alert-frequency
summary-mode summarize
summary-interval 15
summary-key Axxx
specify-global-summary-threshold yes
global-summary-threshold 75
exit
exit
exit
exit
exit
! ------------------------------
service ssh-known-hosts
exit
! ------------------------------
service trusted-certificates
exit
! ------------------------------
service web-server
exit
sensor#
```

Copying and Restoring the Configuration File Using a Remote Server

Use the **copy** [**/erase**] *source_url destination_url keyword* command to copy the configuration file to a remote server. You can then restore the current configuration from the remote server. You are prompted to back up the current configuration first.

Note We recommend copying the current configuration file to a remote server before upgrading.

The following options apply:

- **/erase**—Erases the destination file before copying.

  This keyword only applies to the current-config; the backup-config is always overwritten. If this keyword is specified for destination current-config, the source configuration is applied to the system default configuration. If it is not specified for the destination current-config, the source configuration is merged with the current-config.

- *source_url*—The location of the source file to be copied. It can be a URL or keyword.
- *destination_url*—The location of the destination file to be copied. It can be a URL or a keyword.

The exact format of the source and destination URLs varies according to the file. Here are the valid types:

- ftp:—Source or destination URL for an FTP network server. The syntax for this prefix is:

  ftp:[//[username@] location]/relativeDirectory]/filename

  ftp:[//[username@]location]//absoluteDirectory]/filename
- scp:—Source or destination URL for the SCP network server. The syntax for this prefix is:

  scp:[//[username@] location]/relativeDirectory]/filename

  scp:[//[username@] location]//absoluteDirectory]/filename
- http:—Source URL for the web server. The syntax for this prefix is:

  http:[[/[username@]location]/directory]/filename
- https:—Source URL for the web server. The syntax for this prefix is:

  https:[[/[username@]location]/directory]/filename

Note If you use FTP or SCP protocol, you are prompted for a password.

The following keywords are used to designate the file location on the sensor:

- **current-config**—The current running configuration. The configuration becomes persistent as the commands are entered.
- **backup-config**—The storage location for the configuration backup.

Caution Copying a configuration file from another sensor may result in errors if the sensing interfaces and virtual sensors are not configured the same.

To back up and restore your current configuration, follow these steps:

Step 1 Log in to the CLI using an account with administrator privileges.

Step 2 To back up the current configuration to the remote server:

```
sensor# copy current-config ftp://qa_user@10.89.146.1//tftpboot/update/qmaster89.cfg
Password: ********
```

Step 3 To restore the configuration file that you copied to the remote server:

```
sensor# copy ftp://qa_user@10.89.146.1//tftpboot/update/qmaster89.cfg current-config
Password: ********
Warning: Copying over the current configuration may leave the box in an unstable state.
Would you like to copy current-config to backup-config before proceeding? [yes]:
```

Step 4 Press **Enter** to copy the configuration file or enter **no** to stop.

Creating and Using a Backup Configuration File

To protect your configuration, you can back up the current configuration and then display it to confirm that is the configuration you want to save. If you need to restore this configuration, you can merge the backup configuration file with the current configuration or overwrite the current configuration file with the backup configuration file.

To back up your current configuration, follow these steps:

Step 1 Log in to the CLI using an account with administrator privileges.

Step 2 Save the current configuration:

```
sensor# copy current-config backup-config
```

The current configuration is saved in a backup file.

Step 3 Display the backup configuration file:

```
sensor# more backup-config
```

The backup configuration file is displayed.

Step 4 You can either merge the backup configuration with the current configuration, or you can overwrite the current configuration.

- To merge the backup configuration into the current configuration:

  ```
  sensor# copy backup-config current-config
  ```

- To overwrite the current configuration with the backup configuration:

  ```
  sensor# copy /erase backup-config current-config
  ```

Erasing the Configuration File

Use the **erase** {**backup-config** | **current-config**} command to delete a logical file.

The following options apply:

- **current-config**—The current running configuration. The configuration becomes persistent as the commands are entered.
- **backup-config**—The storage location for the configuration backup.

To erase the current configuration and return all settings back to the default, follow these steps:

Step 1 Log in to the CLI using an account with administrator privileges.

```
sensor# erase current-config
Warning: Removing the current-config file will result in all configuration being reset to
default, including system information such as IP address.
```

```
User accounts will not be erased. They must be removed manually using the "no username"
command.
Continue? []:
```

Step 2 Press **Enter** to continue or enter **no** to stop.

CHAPTER 13

Administrative Tasks for the Sensor

This chapter contains procedures that will help you with the administrative aspects of your sensor. It contains the following sections:

Creating a Banner Login

Use the **banner login** command to create a banner login that will be displayed before the user and password login prompts. The maximum message length is 2500 characters. Use the **no banner login** command to remove the banner.

To create a banner login, follow these steps:

Step 1 Log in to the CLI using an account with administrator privileges.

Step 2 Enter global configuration mode:

```
sensor# configure terminal
```

Step 3 Create the banner login:

```
sensor(config)# banner login
Banner[]:
```

Step 4 Type your message:

```
Banner[]: This message will be displayed on banner login. ^M Thank you
sensor(config)#
```

Note To use a ? or a carriage return in the message, press **Ctrl-V-?** or **Ctrl-V-Enter**. They are represented by ^M.

Example of a completed banner login:

```
This message will be displayed on login.
Thank you
login: cisco
Password:****
```

Step 5 To remove the banner login:

```
sensor(config)# no banner login
```

The banner no longer appears at login.

Terminating CLI Sessions

Use the **clear line** *cli_id* [**message**] command to terminate another CLI session. If you use the **message** keyword, you can send a message along with the termination request to the receiving user. The maximum message length is 2500 characters.

The following options apply:

- *cli_id*—CLI ID number associated with the login session. Use the **show users** command to find the CLI ID number.
- **message**—Message to send to the receiving user.

Caution You can only clear CLI login sessions with the **clear line** command. You cannot clear service logins with this command.

If an administrator tries to log in when the maximum sessions have been reached, the following message appears:

```
Error: The maximum allowed CLI sessions are currently open, would you like to terminate
one of the open sessions? [no]
```

If an operator or viewer tries to log in when the maximum sessions are open, the following message appears:

```
Error: The maximum allowed CLI sessions are currently open, please try again later.
```

To terminate a CLI session, follow these steps:

Step 1 Log in to the CLI using an account with administrator privileges.

Step 2 Find the CLI ID number associated with the login session:

```
sensor# show users
    CLI ID   User    Privilege
*   13533    jtaylor administrator
    15689    jsmith  operator
    20098    viewer  viewer
```

Step 3 Terminate the CLI session of jsmith:

```
sensor# clear line cli_id message
Message[]:
```

Example:

```
sensor# clear line 15689 message
Message{}: Sorry! I need to terminate your session.
sensor#
```

Step 4 The user jsmith receives the following message from the administrator jtaylor:

```
sensor#
***
***
*** Termination request from jtaylor
***
Sorry! I need to terminate your session.
```

Modifying Terminal Properties

Use the **terminal [length]** *screen _length* command to modify terminal properties for a login session. The *screen_ length* option lets you set the number of lines that appear on the screen before the `--more--` prompt is displayed. A value of zero results in no pause in the output. The default value is 24 lines.

Note You are not required to specify the screen length for some types of terminal sessions because the specified screen length can be learned by some remote hosts.

To modify the terminal properties, follow these steps:

Step 1 Log in to the CLI.

Step 2 To have no pause between multi-screen outputs, use 0 for the screen length value:

```
sensor# terminal length 0
```

Note The screen length values are not saved between login sessions.

Step 3 To have the CLI pause and display the `--more--` prompt every 10 lines, use 10 for the *screen length* value:

```
sensor# terminal length 10
```

Events

This section describes how to display and clear events from the Event Store, and contains the following topics:

- Displaying Events, page 13-4
- Clearing Events from the Event Store, page 13-7

Displaying Events

Use the **show events** [{[**alert** [informational] [low] [medium] [high] [**include-traits** ***traits***] [**exclude-traits** ***traits***]] | **error** [warning] [error] [fatal] | **NAC** | **status**}] [*hh:mm:ss* [*month day* [*year*]] | **past** *hh:mm:ss*] command to display events from Event Store.

Events are displayed beginning at the start time. If you do not specify a start time, events are displayed beginning at the current time. If you do not specify an event type, all events are displayed.

Note Events are displayed as a live feed until you cancel the request by pressing Ctrl-C.

The following options apply:

- **alert**—Displays alerts. Provides notification of some suspicious activity that may indicate an attack is in process or has been attempted.

  If no level is selected (informational, low, medium, or high), all alert events are displayed.
- **include-traits**—Displays alerts that have the specified traits.
- **exclude-traits**—Does not display alerts that have the specified traits.
- **traits**—Trait bit position in decimal (0 to 15).
- **error**—Displays error events. Error events are generated by services when error conditions are encountered.
- **NAC**—Displays ARC (block) requests.

  Note ARC is formerly known as Network Access Controller (NAC). This name change has not been completely implemented throughout the IDM and CLI for IPS 5.1.
- **status**—Displays status events.

- **past**—Displays events starting in the past for the specified hours, minutes, and seconds.
- *hh:mm:ss*—Hours, minutes, and seconds in the past to begin the display.

Note The **show events** command waits until a specified event is available. It continues to wait and display events until you exit by pressing Ctrl-C.

To display events from the Event Store, follow these steps:

Step 1 Log in to the CLI.

Step 2 Display all events starting now:

```
sensor#@ show events
evError: eventId=1041472274774840147 severity=warning vendor=Cisco
  originator:
    hostId: sensor2
    appName: cidwebserver
    appInstanceId: 12075
  time: 2003/01/07 04:41:45 2003/01/07 04:41:45 UTC
  errorMessage: name=errWarning received fatal alert: certificate_unknown

evError: eventId=1041472274774840148 severity=error vendor=Cisco
  originator:
    hostId: sensor2
    appName: cidwebserver
    appInstanceId: 351
  time: 2003/01/07 04:41:45 2003/01/07 04:41:45 UTC
  errorMessage: name=errTransport WebSession::sessionTask(6) TLS connection exce
ption: handshake incomplete.
```

The feed continues showing all events until you press **Ctrl-C**.

Step 3 Display the block requests beginning at 10:00 a.m. on February 9, 2005:

```
sensor#@ show events NAC 10:00:00 Feb 9 2005
evShunRqst: eventId=1106837332219222281 vendor=Cisco
  originator:
    deviceName: Sensor1
    appName: NetworkAccessControllerApp
    appInstance: 654
  time: 2005/02/09 10:33:31 2004/08/09 13:13:31
  shunInfo:
    host: connectionShun=false
      srcAddr: 11.0.0.1
      destAddr:
      srcPort:
      destPort:
      protocol: numericType=0 other
    timeoutMinutes: 40
  evAlertRef: hostId=esendHost 123456789012345678
sensor#
```

Step 4 Display errors with the warning level starting at 10:00 a.m. February 9 2005:

```
sensor# show events error warning 10:00:00 Feb 9 2005
evError: eventId=1041472274774840197 severity=warning vendor=Cisco
  originator:
    hostId: sensor
    appName: cidwebserver
    appInstanceId: 12160
```

```
  time: 2003/01/07 04:49:25 2003/01/07 04:49:25 UTC
  errorMessage: name=errWarning received fatal alert: certificate_unknown
```

Step 5 Display alerts from the past 45 seconds:

```
sensor# show events alert past 00:00:45

evIdsAlert: eventId=1109695939102805307 severity=medium vendor=Cisco
  originator:
    hostId: sensor
    appName: sensorApp
    appInstanceId: 367
  time: 2005/03/02 14:15:59 2005/03/02 14:15:59 UTC
  signature: description=Nachi Worm ICMP Echo Request id=2156 version=S54
    subsigId: 0
    sigDetails: Nachi ICMP
  interfaceGroup:
  vlan: 0
  participants:
    attacker:
      addr: locality=OUT 10.89.228.202
    target:
      addr: locality=OUT 10.89.150.185
  riskRatingValue: 70
  interface: fe0_1
  protocol: icmp


evIdsAlert: eventId=1109695939102805308 severity=medium vendor=Cisco
  originator:
--MORE--
```

Step 6 Display events that began 30 seconds in the past:

```
sensor# show events past 00:00:30
evStatus: eventId=1041526834774829055 vendor=Cisco
  originator:
    hostId: sensor
    appName: mainApp
    appInstanceId: 2215
  time: 2003/01/08 02:41:00 2003/01/08 02:41:00 UTC
  controlTransaction: command=getVersion successful=true
    description: Control transaction response.
    requestor:
      user: cids
      application:
        hostId: 64.101.182.101
        appName: -cidcli
        appInstanceId: 2316


evStatus: eventId=1041526834774829056 vendor=Cisco
  originator:
    hostId: sensor
    appName: login(pam_unix)
    appInstanceId: 2315
  time: 2003/01/08 02:41:00 2003/01/08 02:41:00 UTC
  syslogMessage:
    description: session opened for user cisco by cisco(uid=0)
```

Clearing Events from the Event Store

Use the **clear events** command to clear Event Store.

To clear events from Event Store, follow these steps:

Step 1 Log in to the CLI using an account with administrator privileges.

Step 2 Clear Event Store:

```
sensor# clear events
Warning: Executing this command will remove all events currently stored in the event
store.
Continue with clear? []:
```

Step 3 Type **yes** to clear the events.

System Clock

This section explains how to display and manually set the system clock. It contains the following topics:

- Displaying the System Clock, page 13-7
- Manually Setting the Clock, page 13-8

Displaying the System Clock

Use the **show clock** [**detail**] command to display the system clock. You can use the **detail** option to indicate the clock source (NTP or system) and the current summertime setting (if any).

The system clock keeps an authoritative flag that indicates whether the time is authoritative (believed to be accurate). If the system clock has been set by a timing source, such as NTP, the flag is set.

Symbol	Description
*	Time is not authoritative.
(blank)	Time is authoritative.
.	Time is authoritative, but NTP is not synchronized.

To display the system clock, follow these steps:

Step 1 Log in to the CLI.

Step 2 Display the system clock:

```
sensor# show clock
22:39:21 UTC Sat Jan 25 2003
```

Step 3 Display the system clock with details:

```
sensor# show clock detail
22:39:21 CST Sat Jan 25 2003
Time source is NTP
```

```
Summer time starts 02:00:00 CST Sun Apr 7 2004
Summer time ends 02:00:00 CDT Sun Oct 27 2004
```

This indicates that the sensor is getting its time from NTP and that is configured and synchronized.

```
sensor# show clock detail
*12:19:22 CST Sat Dec 04 2004
No time source
Summer time starts 02:00:00 CST Sun Apr 7 2004
Summer time ends 02:00:00 CDT Sun Oct 27 2004
```

This indicates that no time source is configured.

Manually Setting the Clock

Use the **clock set** *hh:mm [:ss] month day year* command to manually set the clock on the appliance. Use this command if no other time sources are available.

Note You do not need to set the system clock if your sensor is synchronized by a valid outside timing mechanism such as an NTP clock source.

For the procedure for configuring NTP, see Configuring NTP, page 4-29. For an explanation of the importance of having a valid time source for the sensor, see Time Sources and the Sensor, page 4-18. For an explanation of what to do if you set the clock incorrectly, see Correcting Time on the Sensor, page 4-21.

The **clock set** command does not apply to the following platforms:

- IDSM-2
- NM-CIDS
- AIP-SSM-10
- AIP-SSM-20

To manually set the clock on the appliance, follow these steps:

Step 1 Log in to the CLI using an account with administrator privileges.

Step 2 Set the clock manually:

```
sensor# clock set 13:21 July 29 2004
```

Note The time format is 24-hour time.

Clearing the Denied Attackers List

Use the **clear denied-attackers** command in service event action rules submode to delete the denied attackers list and clear the virtual sensor statistics.

If your sensor is configured to operate in inline mode, the traffic is passing through the sensor. You can configure signatures to deny packets, connections, and attackers while in inline mode, which means that single packets, connections, and specific attackers will be denied, that is, not transmitted, when the sensor encounters them.

When the signature fires, the attacker is denied and placed in a list. As part of sensor administration, you may want to delete the list or clear the statistics in the list.

To delete the list of denied attackers and clear the statistics, follow these steps:

Step 1 Log in to the CLI using an account with administrator privileges.

Step 2 Display the list of denied IP addresses:

```
sensor# show statistics denied-attackers
Denied Attackers and hit count for each.
   10.20.4.2 = 9
   10.20.5.2 = 5
```

The statistics show that there are two IP addresses being denied at this time.

Step 3 Delete the denied attackers list:

```
sensor# clear denied-attackers
Warning: Executing this command will delete all addresses from the list of
attackers currently being denied by the sensor.
Continue with clear? [yes]:
```

Step 4 Type **yes** to clear the list.

Step 5 Verify that you have cleared the list:

```
sensor# show statistics virtual-sensor
Virtual Sensor Statistics
   Statistics for Virtual Sensor vs0
      Name of current Signature-Definition instance = sig0
      Name of current Event-Action-Rules instance = rules0
      List of interfaces monitored by this virtual sensor = mypair
      Denied Address Information
         Number of Active Denied Attackers = 0
         Number of Denied Attackers Inserted = 2
         Number of Denied Attackers Total Hits = 287
         Number of times max-denied-attackers limited creation of new entry = 0
         Number of exec Clear commands during uptime = 1
      Denied Attackers and hit count for each.
```

There is no longer any information under the `Denied Attackers and hit count for each category`.

Step 6 To clear only the statistics:

```
sensor# show statistics virtual-sensor clear
```

Step 7 Verify that you have cleared the statistics:

```
sensor# show statistics virtual-sensor
Virtual Sensor Statistics
   Statistics for Virtual Sensor vs0
      Name of current Signature-Definition instance = sig0
      Name of current Event-Action-Rules instance = rules0
```

```
      List of interfaces monitored by this virtual sensor = mypair
      Denied Address Information
         Number of Active Denied Attackers = 2
         Number of Denied Attackers Inserted = 0
         Number of Denied Attackers Total Hits = 0
         Number of times max-denied-attackers limited creation of new entry = 0
         Number of exec Clear commands during uptime = 1
      Denied Attackers and hit count for each.
         10.20.2.5 = 0
         10.20.5.2 = 0
```

The statistics have all been cleared except for the `Number of Active Denied Attackers` and `Number of exec Clear commands during uptime` categories. It is important to know if the list has been cleared.

Displaying Statistics

Use the **show statistics virtual-sensor** [**clear**] command to display the statistics for the virtual sensor. Use the **show statistics [analysis-engine | authentication | denied-attackers | event-server | event-store | host | logger | network-access | notification | sdee-server | transaction-server | transaction-source | web-server]** [**clear**] command to generate statistics for each sensor application.

Note The **clear** option is not available for the analysis engine, host, or network access applications.

To display statistics for the sensor, follow these steps:

Step 1 Log in to the CLI.

Step 2 Display the statistics for the virtual sensor:

```
sensor# show statistics virtual-sensor
Virtual Sensor Statistics
   Statistics for Virtual Sensor vs0
      Name of current Signature-Definition instance = sig0
      Name of current Event-Action-Rules instance = rules0
      List of interfaces monitored by this virtual sensor = fe0_1
      General Statistics for this Virtual Sensor
         Number of seconds since a reset of the statistics = 1675
         Measure of the level of resource utilization = 0
         Total packets processed since reset = 241
         Total IP packets processed since reset = 12
         Total packets that were not IP processed since reset = 229
         Total TCP packets processed since reset = 0
         Total UDP packets processed since reset = 0
         Total ICMP packets processed since reset = 12
         Total packets that were not TCP, UDP, or ICMP processed since reset = 0
         Total ARP packets processed since reset = 0
         Total ISL encapsulated packets processed since reset = 0
         Total 802.1q encapsulated packets processed since reset = 0
         Total packets with bad IP checksums processed since reset = 0
         Total packets with bad layer 4 checksums processed since reset = 0
         Total number of bytes processed since reset = 22513
         The rate of packets per second since reset = 0
         The rate of bytes per second since reset = 13
         The average bytes per packet since reset = 93
      Denied Address Information
         Number of Active Denied Attackers = 0
```

```
   Number of Denied Attackers Inserted = 0
   Number of Denied Attackers Total Hits = 0
   Number of times max-denied-attackers limited creation of new entry = 0
   Number of exec Clear commands during uptime = 0
Denied Attackers and hit count for each.
The Signature Database Statistics.
   The Number of each type of node active in the system (can not be reset)
      Total nodes active = 0
      TCP nodes keyed on both IP addresses and both ports = 0
      UDP nodes keyed on both IP addresses and both ports = 0
      IP nodes keyed on both IP addresses = 0
   The number of each type of node inserted since reset
      Total nodes inserted = 28
      TCP nodes keyed on both IP addresses and both ports = 0
      UDP nodes keyed on both IP addresses and both ports = 0
      IP nodes keyed on both IP addresses = 6
   The rate of nodes per second for each time since reset
      Nodes per second = 0
      TCP nodes keyed on both IP addresses and both ports per second = 0
      UDP nodes keyed on both IP addresses and both ports per second = 0
      IP nodes keyed on both IP addresses per second = 0
   The number of root nodes forced to expire because of memory constraints
      TCP nodes keyed on both IP addresses and both ports = 0
Fragment Reassembly Unit Statistics for this Virtual Sensor
   Number of fragments currently in FRU = 0
   Number of datagrams currently in FRU = 0
   Number of fragments received since reset = 0
   Number of fragments forwarded since reset = 0
   Number of fragments dropped since last reset = 0
   Number of fragments modified since last reset = 0
   Number of complete datagrams reassembled since last reset = 0
   Fragments hitting too many fragments condition since last reset = 0
   Number of overlapping fragments since last reset = 0
   Number of Datagrams too big since last reset = 0
   Number of overwriting fragments since last reset = 0
   Number of Initial fragment missing since last reset = 0
   Fragments hitting the max partial dgrams limit since last reset = 0
   Fragments too small since last reset = 0
   Too many fragments per dgram limit since last reset = 0
   Number of datagram reassembly timeout since last reset = 0
   Too many fragments claiming to be the last since last reset = 0
   Fragments with bad fragment flags since last reset = 0
TCP Normalizer stage statistics
   Packets Input = 0
   Packets Modified = 0
   Dropped packets from queue = 0
   Dropped packets due to deny-connection = 0
   Current Streams = 0
   Current Streams Closed = 0
   Current Streams Closing = 0
   Current Streams Embryonic = 0
   Current Streams Established = 0
   Current Streams Denied = 0
Statistics for the TCP Stream Reassembly Unit
   Current Statistics for the TCP Stream Reassembly Unit
      TCP streams currently in the embryonic state = 0
      TCP streams currently in the established state = 0
      TCP streams currently in the closing state = 0
      TCP streams currently in the system = 0
      TCP Packets currently queued for reassembly = 0
   Cumulative Statistics for the TCP Stream Reassembly Unit since reset
      TCP streams that have been tracked since last reset = 0
      TCP streams that had a gap in the sequence jumped = 0
      TCP streams that was abandoned due to a gap in the sequence = 0
```

```
      TCP packets that arrived out of sequence order for their stream = 0
      TCP packets that arrived out of state order for their stream = 0
      The rate of TCP connections tracked per second since reset = 0
SigEvent Preliminary Stage Statistics
   Number of Alerts received = 491
   Number of Alerts Consumed by AlertInterval = 0
   Number of Alerts Consumed by Event Count = 0
   Number of FireOnce First Alerts = 6
   Number of FireOnce Intermediate Alerts = 480
   Number of Summary First Alerts  = 0
   Number of Summary Intermediate Alerts  = 0
   Number of Regular Summary Final Alerts  = 0
   Number of Global Summary Final Alerts  = 0
   Number of Alerts Output for further processing = 491
SigEvent Action Override Stage Statistics
   Number of Alerts received to Action Override Processor = 0
   Number of Alerts where an override was applied = 0
   Actions Added
      deny-attacker-inline = 0
      deny-connection-inline = 0
      deny-packet-inline = 0
      modify-packet-inline = 0
      log-attacker-packets = 0
      log-pair-packets = 0
      log-victim-packets = 0
      produce-alert = 0
      produce-verbose-alert = 0
      request-block-connection = 0
      request-block-host = 0
      request-snmp-trap = 0
      reset-tcp-connection = 0
SigEvent Action Filter Stage Statistics
   Number of Alerts received to Action Filter Processor = 0
   Number of Alerts where an action was filtered = 0
   Number of Filter Line matches = 0
   Actions Filtered
      deny-attacker-inline = 0
      deny-connection-inline = 0
      deny-packet-inline = 0
      modify-packet-inline = 0
      log-attacker-packets = 0
      log-pair-packets = 0
      log-victim-packets = 0
      produce-alert = 0
      produce-verbose-alert = 0
      request-block-connection = 0
      request-block-host = 0
      request-snmp-trap = 0
      reset-tcp-connection = 0
SigEvent Action Handling Stage Statistics.
   Number of Alerts received to Action Handling Processor = 491
   Number of Alerts where produceAlert was forced = 0
   Number of Alerts where produceAlert was off = 0
   Actions Performed
      deny-attacker-inline = 0
      deny-connection-inline = 0
      deny-packet-inline = 0
      modify-packet-inline = 0
      log-attacker-packets = 0
      log-pair-packets = 0
      log-victim-packets = 0
      produce-alert = 11
      produce-verbose-alert = 0
      request-block-connection = 0
```

```
            request-block-host = 5
            request-snmp-trap = 0
            reset-tcp-connection = 0
         Deny Actions Requested in Promiscuous Mode
            deny-packet not performed = 0
            deny-connection not performed = 0
            deny-attacker not performed = 0
            modify-packet not performed = 0
         Number of Alerts where deny-connection was forced for deny-packet action = 0
         Number of Alerts where deny-packet was forced for non-TCP deny-connection action
= 0
      Per-Signature SigEvent count since reset
         Sig 2004 = 5
         Sig 2156 = 486
sensor#
```

Step 3 Display the statistics for AnalysisEngine:

```
sensor# show statistics analysis-engine
Analysis Engine Statistics
   Number of seconds since service started = 1999
   Measure of the level of current resource utilization = 0
   Measure of the level of maximum resource utilization = 0
   The rate of TCP connections tracked per second = 0
   The rate of packets per second = 0
   The rate of bytes per second = 13
   Receiver Statistics
      Total number of packets processed since reset = 290
      Total number of IP packets processed since reset = 12
   Transmitter Statistics
      Total number of packets transmitted = 290
      Total number of packets denied = 0
      Total number of packets reset = 0
   Fragment Reassembly Unit Statistics
      Number of fragments currently in FRU = 0
      Number of datagrams currently in FRU = 0
   TCP Stream Reassembly Unit Statistics
      TCP streams currently in the embryonic state = 0
      TCP streams currently in the established state = 0
      TCP streams currently in the closing state = 0
      TCP streams currently in the system = 0
      TCP Packets currently queued for reassembly = 0
   The Signature Database Statistics.
      Total nodes active = 0
      TCP nodes keyed on both IP addresses and both ports = 0
      UDP nodes keyed on both IP addresses and both ports = 0
      IP nodes keyed on both IP addresses = 0
   Statistics for Signature Events
      Number of SigEvents since reset = 491
   Statistics for Actions executed on a SigEvent
      Number of Alerts written to the IdsEventStore = 11
sensor#
```

Step 4 Display the statistics for authentication:

```
sensor# show statistics authentication
General
   totalAuthenticationAttempts = 2
   failedAuthenticationAttempts = 0
sensor#
```

Step 5 Display the statistics for the denied attackers in the system:

```
sensor# show statistics denied-attackers
Denied Attackers and hit count for each.
```

```
sensor#
```

Step 6 Display the statistics for the event server:

```
sensor# show statistics event-server
General
   openSubscriptions = 0
   blockedSubscriptions = 0
Subscriptions
sensor#
```

Step 7 Display the statistics for Event Store:

```
sensor# show statistics event-store
Event store statistics
   General information about the event store
      The current number of open subscriptions = 2
      The number of events lost by subscriptions and queries = 0
      The number of queries issued = 0
      The number of times the event store circular buffer has wrapped = 0
   Number of events of each type currently stored
      Debug events = 0
      Status events = 9904
      Log transaction events = 0
      Shun request events = 61
      Error events, warning = 67
      Error events, error = 83
      Error events, fatal = 0
      Alert events, informational = 60
      Alert events, low = 1
      Alert events, medium = 60
      Alert events, high = 0
sensor#
```

Step 8 Display the statistics for the host:

```
sensor# show statistics host
General Statistics
   Last Change To Host Config (UTC) = 16:11:05  Thu Feb 10 2005
   Command Control Port Device = FastEthernet0/0
Network Statistics
   fe0_0     Link encap:Ethernet  HWaddr 00:0B:46:53:06:AA
             inet addr:10.89.149.185  Bcast:10.89.149.255  Mask:255.255.255.128
             UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
             RX packets:1001522 errors:0 dropped:0 overruns:0 frame:0
             TX packets:469569 errors:0 dropped:0 overruns:0 carrier:0
             collisions:0 txqueuelen:1000
             RX bytes:57547021 (54.8 MiB)  TX bytes:63832557 (60.8 MiB)
             Interrupt:9 Base address:0xf400 Memory:c0000000-c0000038
NTP Statistics
   status = Not applicable
Memory Usage
   usedBytes = 500592640
   freeBytes = 8855552
   totalBytes = 509448192
Swap Usage
   Used Bytes = 77824
   Free Bytes = 600649728

   Total Bytes = 600727552
CPU Statistics
   Usage over last 5 seconds = 0
   Usage over last minute = 1
   Usage over last 5 minutes = 1
Memory Statistics
```

```
   Memory usage (bytes) = 500498432
   Memory free (bytes) = 894976032
Auto Update Statistics
   lastDirectoryReadAttempt = N/A
   lastDownloadAttempt = N/A
   lastInstallAttempt = N/A
   nextAttempt = N/A
sensor#
```

Step 9 Display the statistics for the logging application:

```
sensor# show statistics logger
The number of Log interprocessor FIFO overruns = 0
The number of syslog messages received = 11
The number of <evError> events written to the event store by severity
   Fatal Severity = 0
   Error Severity = 64
   Warning Severity = 35
   TOTAL = 99
The number of log messages written to the message log by severity
   Fatal Severity = 0
   Error Severity = 64
   Warning Severity = 24
   Timing Severity = 311
   Debug Severity = 31522
   Unknown Severity = 7
   TOTAL = 31928
sensor#
```

Step 10 Display the statistics for ARC:

```
sensor# show statistics network-access
Current Configuration
   LogAllBlockEventsAndSensors = true
   EnableNvramWrite = false
   EnableAclLogging = false
   AllowSensorBlock = false
   BlockMaxEntries = 11
   MaxDeviceInterfaces = 250
   NetDevice
      Type = PIX
      IP = 10.89.150.171
      NATAddr = 0.0.0.0
      Communications = ssh-3des
   NetDevice
      Type = PIX
      IP = 10.89.150.219
      NATAddr = 0.0.0.0
      Communications = ssh-des
   NetDevice
      Type = PIX
      IP = 10.89.150.250
      NATAddr = 0.0.0.0
      Communications = telnet
   NetDevice
      Type = Cisco
      IP = 10.89.150.158
      NATAddr = 0.0.0.0
      Communications = telnet
      BlockInterface
         InterfaceName = ethernet0/1
         InterfaceDirection = out
         InterfacePostBlock = Post_Acl_Test
      BlockInterface
```

```
            InterfaceName = ethernet0/1
            InterfaceDirection = in
            InterfacePreBlock = Pre_Acl_Test
            InterfacePostBlock = Post_Acl_Test
      NetDevice
         Type = CAT6000_VACL
         IP = 10.89.150.138
         NATAddr = 0.0.0.0
         Communications = telnet
         BlockInterface
            InterfaceName = 502
            InterfacePreBlock = Pre_Acl_Test
         BlockInterface
            InterfaceName = 507
            InterfacePostBlock = Post_Acl_Test
   State
      BlockEnable = true
      NetDevice
         IP = 10.89.150.171
         AclSupport = Does not use ACLs
         Version = 6.3
         State = Active
         Firewall-type = PIX
      NetDevice
         IP = 10.89.150.219
         AclSupport = Does not use ACLs
         Version = 7.0
         State = Active
         Firewall-type = ASA
      NetDevice
         IP = 10.89.150.250
         AclSupport = Does not use ACLs
         Version = 2.2
         State = Active
         Firewall-type = FWSM
      NetDevice
         IP = 10.89.150.158
         AclSupport = uses Named ACLs
         Version = 12.2
         State = Active
      NetDevice
         IP = 10.89.150.138
         AclSupport = Uses VACLs
         Version = 8.4
         State = Active
      BlockedAddr
         Host
            IP = 22.33.4.5
            Vlan =
            ActualIp =
            BlockMinutes =
         Host
            IP = 21.21.12.12
            Vlan =
            ActualIp =
            BlockMinutes =
         Host
            IP = 122.122.33.4
            Vlan =
            ActualIp =
            BlockMinutes = 60
            MinutesRemaining = 24
```

```
      Network
         IP = 111.22.0.0
         Mask = 255.255.0.0
         BlockMinutes =
sensor#
```

Step 11 Display the statistics for the notification application:

```
sensor# show statistics notification
General
   Number of SNMP set requests = 0
   Number of SNMP get requests = 0
   Number of error traps sent = 0
   Number of alert traps sent = 0
sensor#
```

Step 12 Display the statistics for the SDEE server:

```
sensor# show statistics sdee-server
General
   Open Subscriptions = 0
   Blocked Subscriptions = 0
   Maximum Available Subscriptions = 5
   Maximum Events Per Retrieval = 500
Subscriptions
sensor#
```

Step 13 Display the statistics for the transaction server:

```
sensor# show statistics transaction-server
General
   totalControlTransactions = 35
   failedControlTransactions = 0
sensor#
```

Step 14 Display the statistics for the transaction source:

```
sensor# show statistics transaction-source
General
   totalControlTransactions = 0
   failedControlTransactions = 0
sensor#
```

Step 15 Display the statistics for Web Server:

```
sensor# show statistics web-server
listener-443
   number of server session requests handled = 61
   number of server session requests rejected = 0
   total HTTP requests handled = 35
   maximum number of session objects allowed = 40
   number of idle allocated session objects = 10
   number of busy allocated session objects = 0
crypto library version = 6.0.3
sensor#
```

Step 16 To clear the statistics for an application, for example, the logging application:

```
sensor# show statistics logger clear
The number of Log interprocessor FIFO overruns = 0
The number of syslog messages received = 141
The number of <evError> events written to the event store by severity
   Fatal Severity = 0
   Error Severity = 14
   Warning Severity = 142
```

```
   TOTAL = 156
The number of log messages written to the message log by severity
   Fatal Severity = 0
   Error Severity = 14
   Warning Severity = 1
   Timing Severity = 0
   Debug Severity = 0
   Unknown Severity = 28
   TOTAL = 43
```

The statistics were retrieved and cleared.

Step 17 Verify that the statistics have been cleared:

```
sensor# show statistics logger
The number of Log interprocessor FIFO overruns = 0
The number of syslog messages received = 0
The number of <evError> events written to the event store by severity
   Fatal Severity = 0
   Error Severity = 0
   Warning Severity = 0
   TOTAL = 0
The number of log messages written to the message log by severity
   Fatal Severity = 0
   Error Severity = 0
   Warning Severity = 0
   Timing Severity = 0
   Debug Severity = 0
   Unknown Severity = 0
   TOTAL = 0
sensor#
```

The statistics all begin from 0.

Displaying Tech Support Information

Use the **show tech-support** [**page**] [**password**] [**destination-url** *destination_url*] command to display system information on the screen or have it sent to a specific URL. You can use the information as a troubleshooting tool with TAC.

The following parameters are optional:

- **page**—Displays the output, one page of information at a time.

  Press **Enter** to display the next line of output or use the spacebar to display the next page of information.
- **password**—Leaves passwords and other security information in the output.
- **destination-url**—Indicates the information should be formatted as HTML and sent to the destination that follows this command. If you use this keyword, the output is not displayed on the screen.
- *destination_url*—Indicates the information should be formatted as HTML. The URL specifies where the information should be sent. If you do not use this keyword, the information is displayed on the screen.

To display tech support information, follow these steps:

Step 1 Log in to the CLI using an account with administrator privileges.

Step 2 View the output on the screen:

```
sensor# show tech-support page
```

The system information appears on the screen, one page at a time. Press the spacebar to view the next page or press **Ctrl-C** to return to the prompt.

Step 3 To send the output (in HTML format) to a file, follow these steps:

a. Type the following command, followed by a valid destination:

```
sensor# show tech-support destination-url destination_url
```

You can specify the following destination types:

- **ftp:**—Destination URL for FTP network server. The syntax for this prefix is `ftp:[[//username@location]/relativeDirectory]/filename` or `ftp:[[//username@location]//absoluteDirectory]/filename`.
- **scp:**—Destination URL for the SCP network server. The syntax for this prefix is `scp:[[//username@]location]/relativeDirectory]/filename` or `scp:[[//username@]location]//absoluteDirectory]/filename`.

For example, to send the tech support output to the file `/absolute/reports/sensor1Report.html`:

```
sensor# show tech support dest
ftp://csidsuser@10.2.1.2//absolute/reports/sensor1Report.html
```

The `password:` prompt appears.

b. Type the password for this user account.

The `Generating report:` message is displayed.

Displaying Version Information

Use the **show version** command to display version information for all installed operating system packages, signature packages, and IPS processes running on the system. To view the configuration for the entire system, use the **more current-config** command.

To display the version and configuration, follow these steps:

Step 1 Log in to the CLI.

Step 2 View version information:

```
sensor# show version
```

The following examples show sample version output for the appliance and the NM-CIDS.

Sample version output for the appliance:

```
sensor# show version
Application Partition:

Cisco Intrusion Prevention System, Version 5.1(0.16)S185.0

OS Version 2.4.26-IDS-smp-bigphys
Platform: IPS-4255-K9
Serial Number: JAB0815R017
No license present
Sensor up-time is 5 days.
Using 722145280 out of 3974291456 bytes of available memory (18% usage)
system is using 17.3M out of 29.0M bytes of available disk space (59% usage)
application-data is using 36.3M out of 166.8M bytes of available disk space (23% usage)
boot is using 39.4M out of 68.6M bytes of available disk space (61% usage)


MainApp          2005_Feb_18_03.00   (Release)   2005-02-18T03:13:47-0600   Running
AnalysisEngine   2005_Feb_18_03.00   (Release)   2005-02-18T03:13:47-0600   Running
CLI              2005_Feb_18_03.00   (Release)   2005-02-18T03:13:47-0600


Upgrade History:

  IDS-K9-min-5.1-0.16 03:00:00 UTC Mon Oct 31 2005

Recovery Partition Version 1.1 - 5.1(0.16)

sensor#
```

Sample version output for NM-CIDS:

```
nm-cids# show version
Application Partition:
Cisco Intrusion Prevention System, Version 5.1(0.16)S185.0

OS Version 2.4.26-IDS-smp-bigphys
Platform: NM-CIDS
Serial Number: JAD06490681
No license present
Sensor up-time is 1 day.
Using 485675008 out of 509448192 bytes of available memory (95% usage)
system is using 17.3M out of 29.0M bytes of available disk space (59% usage)
application-data is using 31.1M out of 166.8M bytes of available disk space (20% usage)
boot is using 39.5M out of 68.6M bytes of available disk space (61% usage)
application-log is using 529.6M out of 2.8G bytes of available disk space (20% usage)


MainApp          2005_Feb_09_03.00   (Release)   2005-02-09T03:22:27-0600   Running
AnalysisEngine   2005_Feb_09_03.00   (Release)   2005-02-09T03:22:27-0600   Running
CLI              2005_Feb_09_03.00   (Release)   2005-02-09T03:22:27-0600


Upgrade History:

  IDS-K9-min-5.1-0.16 03:00:00 UTC Mon Oct 31 2005

Recovery Partition Version 1.1 - 5.1(0.16)

nm-cids#
```

Note If the `--MORE--` prompt is displayed, press the spacebar to see more information or **Ctrl-C** to cancel the output and get back to the CLI prompt.

Step 3 View configuration information:

Note You can use the **more current-config** or **show configuration** commands.

```
sensor# more current-config
! ------------------------------
! Version 5.1(0.16)
! Current configuration last modified Wed Oct 31 03:20:54 2005
! ------------------------------
display-serial
! ------------------------------
service analysis-engine
exit
! ------------------------------
service authentication
exit
! ------------------------------
service event-action-rules rules0
exit
! ------------------------------
service host
network-settings
host-ip 10.89.147.31/25,10.89.147.126
host-name sensor
access-list 0.0.0.0/0
login-banner-text This message will be displayed on banner login.
exit
time-zone-settings
--MORE--
```

Directing Output to a Serial Connection

Use the **display-serial** command to direct all output to a serial connection. This lets you view system messages on a remote console (using the serial port) during the boot process. The local console is not available as long as this option is enabled. Use the **no display-serial** command to reset the output to the local terminal.

Caution If you are connected to the serial port, you will not get any feedback until Linux has fully booted and enabled support for the serial connection.

The **display-serial** command does not apply to the following IPS platforms:

- IDSM-2
- NM-CIDS
- IDS-4215
- IPS-4240

- IPS-4255
- IPS-4260
- AIP-SSM 10
- AIP-SSM 20

To direct output to the serial port, follow these steps:

Step 1 Log in to the CLI using an account with administrator privileges.

Step 2 Direct the output to the serial port:

```
sensor# configure terminal
sensor(config)# display-serial
```

The default is not to direct the output to a serial connection.

Step 3 Reset the output to the local console:

```
sensor(config)# no display-serial
```

Diagnosing Network Connectivity

Use the **ping** *ip_address* [**count**] command to diagnose basic network connectivity.

Caution No command interrupt is available for this command. It must run to completion.

To diagnose basic network connectivity, follow these steps:

Step 1 Log in to the CLI.

Step 2 Ping the address you are interested in:

```
sensor# ping ip_address count
```

The count is the number of echo requests to send. If you do not specify a number, 4 requests are sent. The range is 1 to 10,000.

Example of a successful ping:

```
sensor# ping 10.89.146.110 6
PING 10.89.146.110 (10.89.146.110): 56 data bytes
64 bytes from 10.89.146.110: icmp_seq=0 ttl=61 time=0.3 ms
64 bytes from 10.89.146.110: icmp_seq=1 ttl=61 time=0.1 ms
64 bytes from 10.89.146.110: icmp_seq=2 ttl=61 time=0.1 ms
64 bytes from 10.89.146.110: icmp_seq=3 ttl=61 time=0.2 ms
64 bytes from 10.89.146.110: icmp_seq=4 ttl=61 time=0.2 ms
64 bytes from 10.89.146.110: icmp_seq=5 ttl=61 time=0.2 ms

--- 10.89.146.110 ping statistics ---
6 packets transmitted, 6 packets received, 0% packet loss
round-trip min/avg/max = 0.1/0.1/0.3 ms
```

Example of an unsuccessful ping:

```
sensor# ping 172.21.172.1 3
PING 172.21.172.1 (172.21.172.1): 56 data bytes

--- 172.21.172.1 ping statistics ---
3 packets transmitted, 0 packets received, 100% packet loss
sensor#
```

Resetting the Appliance

Use the **reset** [**powerdown**] command to gracefully shut down the applications running on the appliance and to reboot the appliance. You can include the **powerdown** option to power off the appliance, if possible, or to have the appliance left in a state where the power can be turned off.

Note To reset the modules, see the individual procedures: Resetting IDSM-2, page 15-40, Shutting Down, Reloading, and Resetting NM-CIDS, page 16-7, and Reloading, Shutting Down, Resetting, and Recovering AIP-SSM, page 14-5.

Shut down (stopping the applications) begins immediately after you execute the command. Shutdown can take a while, and you can still access CLI commands while it is taking place, but the session will be terminated without warning.

To reset the appliance, follow these steps:

Step 1 Log in to the CLI using an account with administrator privileges.

Step 2 To stop all applications and reboot the appliance, follow these steps. Otherwise, to power down the appliance, go to Step 4.

```
sensor# reset
Warning: Executing this command will stop all applications and reboot the node.
Continue with reset? []:
```

Step 3 Type **yes** to continue the reset:

```
sensor# yes
Request Succeeded.
sensor#
```

Step 4 To stop all applications and power down the appliance:

```
sensor# reset powerdown
Warning: Executing this command will stop all applications and power off the node if
possible. If the node can not be powered off it will be left in a state that is safe to
manually power down.
Continue with reset? []:
```

Step 5 Type **yes** to continue with the reset and powerdown:

```
sensor# yes
Request Succeeded.
sensor#
```

Displaying Command History

Use the **show history** command to obtain a list of the commands you have entered in the current menu. The maximum number of commands in the list is 50.

To obtain a list of the commands you have used recently, follow these steps:

Step 1 Log in to the CLI.

Step 2 Show the history of the commands you have used in EXEC mode:

```
sensor# show history
clear line
configure terminal
show history
```

Step 3 Show the history of the commands you have used in network access mode:

```
sensor# configure terminal
sensor (config)# service network-access
sensor (config-net)# show history
show settings
show settings terse
show settings | include profile-name|ip-address
exit
show history
sensor (config-net)#
```

Displaying Hardware Inventory

Use the **show inventory** command to display PEP information. This command displays the UDI information that consists of the PID, the VID, and the SN of your sensor.

PEP information provides an easy way to obtain the hardware version and serial number through the CLI.

The **show inventory** command does not apply to the following platforms:

- IDSM-2
- NM-CIDS
- IDS-4210
- IDS-4215

- IDS-4235
- IDS-4250

To display PEP information, follow these steps:

Step 1 Log in to the CLI.

Step 2 Display the PEP information:

```
sensor# show inventory

Name: "Chassis", DESCR: "IPS 4255 Intrusion Prevention Sensor"
PID: IPS-4255-K9, VID: V01 , SN: JAB0815R017

Name: "Power Supply", DESCR: ""
PID: ASA-180W-PWR-AC, VID: V01 , SN: 123456789AB
sensor#

sensor# show inventory

Name: "Module", DESCR: "ASA 5500 Series Security Services Module-20"
PID: ASA-SSM-20, VID: V01 , SN: JAB0815R036
sensor#

sensor-4240# show inventory

Name: "Chassis", DESCR: "IPS 4240 Appliance Sensor"
PID: IPS-4240-K9, VID: V01 , SN: P3000000653
sensor-4240#
```

You can use this information when dealing with the TAC.

Tracing the Route of an IP Packet

Use the **trace** *ip_address count* command to display the route an IP packet takes to a destination. The *ip_address* option is the address of the system to trace the route to. The *count* option lets you define how many hops you want to take. The default is 4. The valid values are 1 to 256.

Caution There is no command interrupt available for this command. It must run to completion.

To trace the route of an IP packet, follow these steps:

Step 1 Log in to the CLI.

Step 2 Display the route of IP packet you are interested in:

```
sensor# trace 10.1.1.1
traceroute to 10.1.1.1 (10.1.1.1), 4 hops max, 40 byte packets
 1  10.89.130.1 (10.89.130.1)  0.267 ms  0.262 ms  0.236 ms
 2  10.89.128.17 (10.89.128.17)  0.24 ms *  0.399 ms
 3  * 10.89.128.17 (10.89.128.17)  0.424 ms *
 4  10.89.128.17 (10.89.128.17)  0.408 ms *  0.406 ms
sensor#
```

Step 3 To have the route take more hops than the default of 4, use the *count* option:

```
sensor# trace 10.1.1.1 8
traceroute to 10.1.1.1 (10.1.1.1), 8 hops max, 40 byte packets
 1  10.89.130.1 (10.89.130.1)  0.35 ms  0.261 ms  0.238 ms
 2  10.89.128.17 (10.89.128.17)  0.36 ms *  0.344 ms
 3  * 10.89.128.17 (10.89.128.17)  0.465 ms *
 4  10.89.128.17 (10.89.128.17)  0.319 ms *  0.442 ms
 5  * 10.89.128.17 (10.89.128.17)  0.304 ms *
 6  10.89.128.17 (10.89.128.17)  0.527 ms *  0.402 ms
 7  * 10.89.128.17 (10.89.128.17)  0.39 ms *
 8  10.89.128.17 (10.89.128.17)  0.37 ms *  0.486 ms
sensor#
```

Displaying Submode Settings

Use the **show settings** [**terse**] command in any submode to view the contents of the current configuration.

To display the current configuration settings for a submode, follow these steps:

Step 1 Log in to the CLI.

Step 2 Show the current configuration for ARC submode:

```
sensor# configure terminal
sensor (config)# service network-access
sensor (config-net)# show settings
   general
   ----------------------------------------------
      log-all-block-events-and-errors: true <defaulted>
      enable-nvram-write: false <defaulted>
      enable-acl-logging: false <defaulted>
      allow-sensor-block: false <defaulted>
      block-enable: true <defaulted>
      block-max-entries: 250 <defaulted>
      max-interfaces: 250 default: 250
      master-blocking-sensors (min: 0, max: 100, current: 0)
      ----------------------------------------------
      ----------------------------------------------
      never-block-hosts (min: 0, max: 250, current: 0)
      ----------------------------------------------
      ----------------------------------------------
      never-block-networks (min: 0, max: 250, current: 0)
      ----------------------------------------------
      ----------------------------------------------
      block-hosts (min: 0, max: 250, current: 0)
      ----------------------------------------------
      ----------------------------------------------
      block-networks (min: 0, max: 250, current: 0)
      ----------------------------------------------
      ----------------------------------------------
   ---------------------------------------------
   user-profiles (min: 0, max: 250, current: 11)
   ---------------------------------------------
      profile-name: 2admin
      ----------------------------------------------
         enable-password: <hidden>
         password: <hidden>
```

```
      username: pix default:
   ----------------------------------------------
   profile-name: r7200
   ----------------------------------------------
      enable-password: <hidden>
      password: <hidden>
      username: netrangr default:
   ----------------------------------------------
   profile-name: insidePix
   ----------------------------------------------
      enable-password: <hidden>
      password: <hidden>
      username:  <defaulted>
   ----------------------------------------------
   profile-name: qatest
   ----------------------------------------------
      enable-password: <hidden>
      password: <hidden>
      username:  <defaulted>
   ----------------------------------------------
   profile-name: fwsm
   ----------------------------------------------
      enable-password: <hidden>
      password: <hidden>
      username: pix default:
   ----------------------------------------------
   profile-name: outsidePix
   ----------------------------------------------
      enable-password: <hidden>
      password: <hidden>
      username: pix default:
   ----------------------------------------------
   profile-name: cat
   ----------------------------------------------
      enable-password: <hidden>
      password: <hidden>
      username:  <defaulted>
   ----------------------------------------------
   profile-name: rcat
   ----------------------------------------------
      enable-password: <hidden>
      password: <hidden>
      username: cisco default:
   ----------------------------------------------
   profile-name: nopass
   ----------------------------------------------
      enable-password: <hidden>
      password: <hidden>
      username:  <defaulted>
   ----------------------------------------------
   profile-name: test
   ----------------------------------------------
      enable-password: <hidden>
      password: <hidden>
      username: pix default:
   ----------------------------------------------
   profile-name: sshswitch
   ----------------------------------------------
      enable-password: <hidden>
      password: <hidden>
      username: cisco default:
   ----------------------------------------------
-----------------------------------------------
cat6k-devices (min: 0, max: 250, current: 1)
```

```
      ----------------------------------------------
         ip-address: 10.89.147.61
         ---------------------------------------------
            communication: telnet default: ssh-3des
            nat-address: 0.0.0.0 <defaulted>
            profile-name: cat
            block-vlans (min: 0, max: 100, current: 1)
            ----------------------------------------------
               vlan: 1
               ----------------------------------------------
                  pre-vacl-name:  <defaulted>
                  post-vacl-name:  <defaulted>
               ----------------------------------------------
            ----------------------------------------------
         ---------------------------------------------
      ----------------------------------------------
      router-devices (min: 0, max: 250, current: 1)
      ----------------------------------------------
         ip-address: 10.89.147.54
         ---------------------------------------------
            communication: telnet default: ssh-3des
            nat-address: 0.0.0.0 <defaulted>
            profile-name: r7200
            block-interfaces (min: 0, max: 100, current: 1)
            ----------------------------------------------
               interface-name: fa0/0
               direction: in
               ----------------------------------------------
                  pre-acl-name:  <defaulted>
                  post-acl-name:  <defaulted>
               ----------------------------------------------
            ----------------------------------------------
         ---------------------------------------------
      ----------------------------------------------
      firewall-devices (min: 0, max: 250, current: 2)
      ----------------------------------------------
         ip-address: 10.89.147.10
         ---------------------------------------------
            communication: telnet default: ssh-3des
            nat-address: 0.0.0.0 <defaulted>
            profile-name: insidePix
         ---------------------------------------------
         ip-address: 10.89.147.82
         ---------------------------------------------
            communication: ssh-3des <defaulted>
            nat-address: 0.0.0.0 <defaulted>
            profile-name: f1
         ---------------------------------------------
      ----------------------------------------------
   sensor (config-net)#
```

Step 3 Show the ARC settings in terse mode:

```
sensor(config-net)# show settings terse
   general
   ----------------------------------------------
      log-all-block-events-and-errors: true <defaulted>
      enable-nvram-write: false <defaulted>
      enable-acl-logging: false <defaulted>
      allow-sensor-block: false <defaulted>
      block-enable: true <defaulted>
      block-max-entries: 250 <defaulted>
      max-interfaces: 250 default: 250
      master-blocking-sensors (min: 0, max: 100, current: 0)
```

```
      ----------------------------------------------
      ----------------------------------------------
      never-block-hosts (min: 0, max: 250, current: 0)
      ----------------------------------------------
      ----------------------------------------------
      never-block-networks (min: 0, max: 250, current: 0)
      ----------------------------------------------
      ----------------------------------------------
      block-hosts (min: 0, max: 250, current: 0)
      ----------------------------------------------
      ----------------------------------------------
      block-networks (min: 0, max: 250, current: 0)
      ----------------------------------------------
      ----------------------------------------------
   ----------------------------------------------
   user-profiles (min: 0, max: 250, current: 11)
   ----------------------------------------------
      profile-name: 2admin
      profile-name: r7200
      profile-name: insidePix
      profile-name: qatest
      profile-name: fwsm
      profile-name: outsidePix
      profile-name: cat
      profile-name: rcat
      profile-name: nopass
      profile-name: test
      profile-name: sshswitch
   ----------------------------------------------
   cat6k-devices (min: 0, max: 250, current: 1)
   ----------------------------------------------
      ip-address: 10.89.147.61
   ----------------------------------------------
   router-devices (min: 0, max: 250, current: 1)
   ----------------------------------------------
      ip-address: 10.89.147.54
   ----------------------------------------------
   firewall-devices (min: 0, max: 250, current: 2)
   ----------------------------------------------
      ip-address: 10.89.147.10
      ip-address: 10.89.147.82
   ----------------------------------------------
sensor(config-net)#
```

Step 4 You can use the **include** keyword to show settings in a filtered output, for example, to show only profile names and IP addresses in the ARC configuration:

```
sensor(config-net)# show settings | include profile-name|ip-address
      profile-name: 2admin
      profile-name: r7200
      profile-name: insidePix
      profile-name: qatest
      profile-name: fwsm
      profile-name: outsidePix
      profile-name: cat
      profile-name: rcat
      profile-name: nopass
      profile-name: test
      profile-name: sshswitch
```

```
      ip-address: 10.89.147.61
         profile-name: cat
      ip-address: 10.89.147.54
         profile-name: r7200
      ip-address: 10.89.147.10
         profile-name: insidePix
      ip-address: 10.89.147.82
         profile-name: test
sensor(config-net)#
```

CHAPTER 14

Configuring AIP-SSM

This chapter contains procedures that are specific to configuring AIP-SSM. It contains the following sections:

Configuration Sequence

Perform the following tasks to configure AIP-SSM:

1. Log in to AIP-SSM.

   For the procedure, see Logging In to AIP-SSM, page 2-7.
2. Initialize AIP-SSM.

   Run the **setup** command to initialize AIP-SSM.

   For the procedure, see Initializing the Sensor, page 3-2.
3. Verify the AIP-SSM initialization.

   For the procedure, see Verifying AIP-SSM Initialization, page 14-2.
4. Configure ASA to send IPS traffic to AIP-SSM.

   For the procedure, see Sending Traffic to AIP-SSM, page 14-2.
5. Perform other initial tasks, such as adding users, trusted hosts, and so forth.

   For the procedures, see Chapter 4, "Initial Configuration Tasks."
6. Configure intrusion prevention.

   For the procedures, see Chapter 6, "Configuring Event Action Rules," Chapter 7, "Defining Signatures," and Chapter 10, "Configuring Attack Response Controller for Blocking and Rate Limiting."
7. Perform miscellaneous tasks to keep your AIP-SSM running smoothly.

   For the procedures, see Chapter 13, "Administrative Tasks for the Sensor."

8. Upgrade the IPS software with new signature updates and service packs.

   For more information, see Chapter 18, "Obtaining Software."

9. Reimage AIP-SSM when needed.

   For the procedure, see Installing the AIP-SSM System Image, page 17-38.

Verifying AIP-SSM Initialization

You can use the **show module** *slot* **details** command to verify that you have initialized AIP-SSM and to verify that you have the correct software version.

To verify initialization, follow these steps:

Step 1 Log in to ASA.

Step 2 Obtain the details about AIP-SSM:

```
asa# show module 1 details
Getting details from the Service Module, please wait...
ASA 5500 Series Security Services Module-20
Model:              ASA-SSM-20
Hardware version:   0.2
Serial Number:      P2B000005D0
Firmware version:   1.0(10)0
Software version:   5.1(0.05)S179.0
Status:             Up
Mgmt IP addr:       10.89.149.219
Mgmt web ports:     443
Mgmt TLS enabled:   false
asa#
```

Step 3 Confirm the information. If you need to change anything, see Configuration Sequence, page 14-1.

Sending Traffic to AIP-SSM

This section describes how to configure AIP-SSM to receive IPS traffic from ASA (inline or promiscuous mode), and contains the following sections:

- Overview, page 14-2
- Configuring ASA to Send IPS Traffic to AIP-SSM, page 14-3

Overview

ASA diverts packets to AIP-SSM just before the packet exits the egress interface (or before VPN encryption occurs, if configured) and after other firewall policies are applied. For example, packets that are blocked by an access list are not forwarded to AIP-SSM. You can configure AIP-SSM to inspect traffic in inline or promiscuous mode and in fail-open or fail-over mode.

On ASA, to identify traffic to be diverted to and inspected by AIP-SSM:

1. Create or use an existing ACL.
2. Use the **class-map** command to define the IPS traffic class.
3. Use the **policy-map** command to create an IPS policy map by associating the traffic class with one or more actions.
4. Use the **service-policy** command to create an IPS security policy by associating the policy map with one or more interfaces.

   You can use the ASA CLI or ASDM to configure IPS traffic inspection.

Configuring ASA to Send IPS Traffic to AIP-SSM

Note For more information on these commands, refer to Chapter 18, "Using Modular Policy Framework," in *Cisco Security Appliance Command Line Configuration Guide*.

The following options apply:

- **access-list** *word*—Configures an access control element; *word* is the access list identifier (up to 241 characters).
- **class-map** *class_map_name*—Defines the IPS traffic class.
- **match**—Identifies the traffic included in the traffic class.

  A traffic class map contains a **match** command. When a packet is matched against a class map, the match result is either a match or a no match.

  - **access-list**—Matches an access list.
  - **any**—Matches any packet.
- **policy-map** *policy_map_name*—Creates an IPS policy map by associating the traffic class with one or more actions.
- **ips {inline | promiscuous][fail-close | fail-open}**—Assigns traffic to AIP-SSM:
  - **inline**—Places AIP-SSM directly in the traffic flow.

    No traffic can continue through ASA without first passing through and being inspected by AIP-SSM. This mode is the most secure because every packet is analyzed before being permitted through. Also, AIP-SSM can implement a blocking policy on a packet-by-packet basis. This mode, however, can affect throughput.
  - **promiscuous**—Sends a duplicate stream of traffic to AIP-SSM.

    This mode is less secure, but has little impact on traffic throughput. Unlike when in inline mode, AIP-SSM cannot block traffic by instructing ASA to block the traffic or by resetting a connection on ASA.
  - **fail-close**—Sets ASA to block all traffic if AIP-SSM is unavailable.
  - **fail-open**—Sets ASA to permit all traffic through, uninspected, if AIP-SSM is unavailable.

Note ASA fail-open/fail-close behavior depends on low-level heartbeats, which are turned off when AIP-SSM is shut down or reset. If AIP-SSM fails, ASA cannot detect this failure because the heartbeats are still received. For inline inspection of traffic, use IPS bypass mode to drop or permit traffic through. For more information on bypass mode, see ASA, AIP-SSM, and Bypass Mode, page 14-5.

- **service-policy** *service_policy_name* [**global** | **interface** *interface_name*]—Creates an IPS security policy by associating the policy map with one or more interfaces.
  - **global**—Applies the policy map to all interfaces.

    Only one global policy is allowed. You can override the global policy on an interface by applying a service policy to that interface. You can only apply one policy map to each interface.
  - **interface**—Applies the policy to one interface.

    You can assign a different policy for each interface.

To send traffic from ASA to AIP-SSM for the IPS to inspect, follow these steps:

Step 1 Log in to ASA.

Step 2 Enter configuration mode:

```
asa# configure terminal
```

Step 3 Create an IPS access list:

```
asa(config)# access-list IPS permit ip any any
```

Step 4 Define the IPS traffic class:

```
asa(config)# class-map class_map_name
asa(config-cmap)# match {access-list | any}
```

Step 5 Define the IPS policy map:

```
asa(config-cmap)# policy-map policy_map_name
```

Step 6 Identify the class map from Step 5 to which you want to assign an action:

```
asa(config-pmap)# class class_map_name
```

Step 7 Assign traffic to AIP-SSM:

```
asa(config-pmap-c)# ips {inline | promiscuous} [fail-close | fail-open}
```

Step 8 Define the IPS service policy:

```
asa(config-pmap-c)# service-policy policymap_name {global | interface interface_name}
```

Step 9 Verify the settings:

```
asa(config-pmap-c)# show running-config
!
class-map my_ips_class
match access-list IPS
class-map all_traffic
 match access-list all_traffic
class-map inspection_default
 match default-inspection-traffic
!
!
```

```
policy-map my-ids-policy
 class my-ips-class
  ips promiscuous fail-close
!
service-policy my-ids-policy global
```

Step 10 Exit and save the configuration:

```
asa(config-pmap-c)# exit
asa(config-pmap)# exit
asa(config)# exit
asa#
```

The following example diverts all IP traffic to AIP-SSM in inline mode, and blocks all IP traffic should AIP-SSM fail for any reason:

```
hostname(config)# access-list IPS permit ip any any
hostname(config)# class-map my-ips-class
hostname(config-cmap)# match access-list IPS
hostname(config-cmap)# policy-map my-ids-policy
hostname(config-pmap)# class my-ips-class
hostname(config-pmap-c)# ips inline fail-close
hostname(config-pmap-c)# service-policy my-ids-policy global
```

ASA, AIP-SSM, and Bypass Mode

The following conditions apply to bypass mode and AIP-SSM:

- Bypass Auto or Off

  ASA permits or blocks traffic from going through according to the configured fail-open or fail-close rules when AIP-SSM is shut down or reset.

- Bypass Auto

  If SensorApp stops in AIP-SSM, ASA permits all traffic through regardless of the configured fail-open or fail-close rules, because the AIP-SSM NIC driver is still functioning and passing heartbeat packets.

- Bypass Off

  If SensorApp stops in AIP-SSM, ASA stops all traffic from going through regardless of the configured fail-open or fail-close rules.

For more information on IPS software bypass mode, see Inline Bypass Mode, page 5-22.

Reloading, Shutting Down, Resetting, and Recovering AIP-SSM

Note You can enter the **hw-module** commands from privileged EXEC mode or from global configuration mode. You can enter the commands in single routed mode and single transparent mode. For adaptive security devices operating in multi-mode (routed or transparent multi-mode) you can only execute the **hw-module** commands from the system context (not from administrator or user contexts).

Use the following commands to reload, shut down, reset, and recover AIP-SSM directly from ASA:

- **hw-module module 1 reload**

  This command reloads the software on AIP-SSM without doing a hardware reset. It is effective only when AIP-SSM is in the Up state.

- **hw-module module 1 shutdown**

  This command shuts down the software on AIP-SSM. It is effective only when AIP-SSM is in Up state.

- **hw-module module 1 reset**

  This command performs a hardware reset of AIP-SSM. It is applicable when the card is in the Up/Down/Unresponsive/Recover states.

- **hw-module module 1 recover** [**boot** | **stop** | **configure**]

  The **recover** command displays a set of interactive options for setting or changing the recovery parameters. You can change the parameter or keep the existing setting by pressing Enter.

  For the procedure for recovering AIP-SSM, see Installing the AIP-SSM System Image, page 17-38.

  – **hw-module module 1 recover boot**

    This command initiates recovery of AIP-SSM. It is applicable only when AIP-SSM is in the Up state.

  – **hw-module module 1 recover stop**

    This command stops recovery of AIP-SSM. It is applicable only when AIP-SSM is in the Recover state.

Caution If AIP-SSM recovery needs to be stopped, you must issue the **hw-module module 1 recover stop** command within 30 to 45 seconds after starting AIP-SSM recovery. Waiting any longer can lead to unexpected consequences. For example, AIP-SSM may come up in the Unresponsive state.

  – **hw-module module 1 recover configure**

    Use this command to configure parameters for module recovery. The essential parameters are the IP address and recovery image TFTP URL location.

    Example:

```
aip-ssm# hardware-module module 1 recover configure
Image URL [tftp://10.89.146.1/IPS-SSM-K9-sys-1.1-a-5.1-1.img]:
Port IP Address [10.89.149.226]:
VLAN ID [0]:
Gateway IP Address [10.89.149.254]: ì
```

CHAPTER 15

Configuring IDSM-2

Note Catalyst 6500 Series Switch is used generically to refer to both the 6500 series switches and the 7600 series routers.

This chapter contains procedures that are specific to configuring IDSM-2. Once you set up IDSM-2 to receive traffic from the network, you can configure it for intrusion prevention. It contains the following sections:

Configuration Sequence

Perform the following tasks to configure IDSM-2:

1. Configure the Catalyst 6500 series switch for command and control access to IDSM-2.

   For the procedure, see Configuring the Catalyst 6500 Series Switch for Command and Control Access to IDSM-2, page 15-5.

2. Log in to IDSM-2.

   For the procedure to session to the IDSM-2, see Logging In to IDSM-2, page 2-4.

3. Configure the switch to send traffic to be monitored to IDSM-2.
   - Configuring the Catalyst Series 6500 Switch for IDSM-2 in Promiscuous Mode, page 15-8
   - Configuring the Catalyst Series 6500 Switch for IDSM-2 in Inline Mode, page 15-19
   - Configuring the Catalyst Series 6500 Switch for IDSM-2 for Inline VLAN Pair Mode, page 15-21
   - Configuring EtherChannel Load Balancing, page 15-24
4. Initialize IDSM-2.

   Run the **setup** command to initialize IDSM-2. During setup, you can configure the interfaces of IDSM-2. If you need to change the interface configuration, see Chapter 5, "Configuring Interfaces."

   For the procedure, see Initializing the Sensor, page 3-2.
5. Create the service account.

   For the procedure, see Creating the Service Account, page 4-13.
6. Perform the other initial tasks, such as adding users, trusted hosts, and so forth.

   For the procedures, see Chapter 4, "Initial Configuration Tasks."
7. Configure intrusion prevention.

   For the procedures, see Chapter 6, "Configuring Event Action Rules," Chapter 7, "Defining Signatures," and Chapter 10, "Configuring Attack Response Controller for Blocking and Rate Limiting."
8. Perform miscellaneous tasks to keep IDSM-2 running smoothly.

   For the procedures, see Chapter 13, "Administrative Tasks for the Sensor" and Administrative Tasks for IDSM-2, page 15-38.
9. Upgrade the IPS software with new signature updates and service packs.

   See Chapter 18, "Obtaining Software" for more information.
10. Reimage the application partition and the maintenance partition when needed.

    For the procedures, see Installing the IDSM-2 System Image, page 17-27.

Verifying IDSM-2 Installation

Use the **show module** command to verify that the switch acknowledges IDSM-2 and has brought it online.

To verify the installation, follow these steps:

Step 1 Log in to the console.

Step 2 For Catalyst software:

```
console> (enable) show module
Mod Slot Ports Module-Type                Model                Sub Status
--- ---- ----- ------------------------- -------------------- --- --------
1   1    2     1000BaseX Supervisor      WS-X6K-SUP1A-2GE     yes ok
15  1    1     Multilayer Switch Feature WS-F6K-MSFC          no  ok
2   2    48    10/100BaseTX Ethernet     WS-X6248-RJ-45       no  ok
3   3    48    10/100/1000BaseT Ethernet WS-X6548-GE-TX       no  ok
4   4    16    1000BaseX Ethernet        WS-X6516A-GBIC       no  ok
6   6    8     Intrusion Detection Mod   WS-SVC-IDSM2         yes ok
```

```
Mod Module-Name          Serial-Num
--- -------------------- -----------
1                        SAD041308AN
15                       SAD04120BRB
2                        SAD03475400
3                        SAD073906RC
4                        SAL0751QYN0
6                        SAD062004LV

Mod MAC-Address(es)                         Hw     Fw          Sw
--- -------------------------------------- ------ ---------- -----------------
1   00-d0-c0-cc-0e-d2 to 00-d0-c0-cc-0e-d3 3.1    5.3.1      8.4(1)
    00-d0-c0-cc-0e-d0 to 00-d0-c0-cc-0e-d1
    00-30-71-34-10-00 to 00-30-71-34-13-ff
15  00-30-7b-91-77-b0 to 00-30-7b-91-77-ef 1.4    12.1(23)E2 12.1(23)E2
2   00-30-96-2b-c7-2c to 00-30-96-2b-c7-5b 1.1    4.2(0.24)V 8.4(1)
3   00-0d-29-f6-01-98 to 00-0d-29-f6-01-c7 5.0    7.2(1)     8.4(1)
4   00-0e-83-af-15-48 to 00-0e-83-af-15-57 1.0    7.2(1)     8.4(1)
6   00-e0-b0-ff-3b-80 to 00-e0-b0-ff-3b-87 0.102  7.2(0.67)  5.0(0.30)

Mod Sub-Type                Sub-Model           Sub-Serial  Sub-Hw Sub-Sw
--- ----------------------- ------------------- ----------- ------ ------
1   L3 Switching Engine     WS-F6K-PFC          SAD041303G6 1.1
6   IDS 2 accelerator board WS-SVC-IDSUPG       .           2.0
console> (enable)
```

Step 3 For Cisco IOS software:

```
router# show module
Mod Ports Card Type                                Model              Serial No.
--- ----- ------------------------------------- ------------------ -----------
  1   48  48 port 10/100 mb RJ-45 ethernet       WS-X6248-RJ-45     SAD0401012S
  2   48  48 port 10/100 mb RJ45                 WS-X6348-RJ-45     SAL04483QBL
  3   48  SFM-capable 48 port 10/100/1000mb RJ45 WS-X6548-GE-TX     SAD073906GH
  6   16  SFM-capable 16 port 1000mb GBIC        WS-X6516A-GBIC     SAL0740MMYJ
  7    2  Supervisor Engine 720 (Active)         WS-SUP720-3BXL     SAD08320L2T
  9    1  1 port 10-Gigabit Ethernet Module      WS-X6502-10GE      SAD071903BT
 10    3  Anomaly Detector Module                WS-SVC-ADM-1-K9    SAD084104JR
 11    8  Intrusion Detection System             WS-SVC-IDSM2       SAD05380608
 13    8  Intrusion Detection System             WS-SVC-IDSM-2      SAD072405D8

Mod MAC addresses                       Hw    Fw           Sw           Status
--- ---------------------------------- ------ ------------ ------------ -------
  1  00d0.d328.e2ac to 00d0.d328.e2db   1.1   4.2(0.24)VAI 8.5(0.46)ROC Ok
  2  0003.6c14.e1d0 to 0003.6c14.e1ff   1.4   5.4(2)       8.5(0.46)ROC Ok
  3  000d.29f6.7a80 to 000d.29f6.7aaf   5.0   7.2(1)       8.5(0.46)ROC Ok
  6  000d.ed23.1658 to 000d.ed23.1667   1.0   7.2(1)       8.5(0.46)ROC Ok
  7  0011.21a1.1398 to 0011.21a1.139b   4.0   8.1(3)       12.2(PIKESPE Ok
  9  000d.29c1.41bc to 000d.29c1.41bc   1.3   Unknown      Unknown      PwrDown
 10  000b.fcf8.2ca8 to 000b.fcf8.2caf   0.101 7.2(1)       4.0(0.25)    Ok
 11  00e0.b0ff.3340 to 00e0.b0ff.3347   0.102 7.2(0.67)    5.0(1)       Ok
 13  0003.feab.c850 to 0003.feab.c857   4.0   7.2(1)       5.0(1)       Ok

Mod Sub-Module                  Model               Serial        Hw     Status
--- --------------------------- ------------------ ------------ ------- -------
  7 Policy Feature Card 3       WS-F6K-PFC3BXL     SAD083305A1   1.3    Ok
  7 MSFC3 Daughterboard         WS-SUP720          SAD083206JX   2.1    Ok
 11 IDS 2 accelerator board     WS-SVC-IDSUPG      .             2.0    Ok
 13 IDS 2 accelerator board     WS-SVC-IDSUPG      0347331976    2.0    Ok

Mod Online Diag Status
--- -------------------
  1 Pass
```

```
  2 Pass
  3 Pass
  6 Pass
  7 Pass
  9 Unknown
 10 Not Applicable
 11 Pass
 13 Pass
router#
```

Note It is normal for the status to read `other` when IDSM-2 is first installed. After IDSM-2 completes the diagnostics routines and comes online, the status reads `ok`. Allow up to 5 minutes for IDSM-2 to come online.

For information on enabling a full memory test after verifying IDSM-2 installation, see Enabling Full Memory Tests, page 15-38.

Minimum Supported IDSM-2 Configurations

Note The following matrix is not intended to recommend any particular version, but rather lists the earliest supported versions.

Table 15-1 lists the minimum supported configurations for IDSM-2.

Table 15-1 Minimum Catalyst 6500 Software Version for IDSM-2 Feature Support

Catalyst/IDSM-2 Feature	Catalyst Software				Cisco IOS Software			
	Sup1	Sup2	Sup32	Sup720	Sup1	Sup2	Sup32	Sup720
SPAN	7.5(1)	7.5(1)	8.4(1)	8.1(1)	12.1(19)E1	12.1(19)E1 12.2(18)SXF1	12.2(18)SXF1	12.2(14)SX1
VACL capture[1]	7.5(1)	7.5(1)	8.4(1)	8.1(1)	12.1(19)E1	12.1(19)E1 12.2(18)SXF1	12.2(18)SXF1	12.2(14)SX1
ECLB with VACL capture[2]	8.5(1)	8.5(1)	8.5(1)	8.5(1)	N/A	12.2(18)SXF4	12.2(18)SXF1	12.2(18)SXE1
Inline interface pairs	8.4(1)	8.4(1)	8.4(1)	8.4(1)	N/A	12.2(18)SXF4	12.2(18)SXF4	12.2(18)SXE1
ECLB with inline interface pairs	8.5(1)	8.5(1)	8.5(1)	8.5(1)	N/A	12.2(18)SXF4	12.2(18)SXF4	12.2(18)SXF4
Inline VLAN pairs	8.4(1)	8.4(1)	8.4(1)	8.4(1)	N/A	12.2(18)SXF4	12.2(18)SXF4	12.2(18)SXF4
ECLB with inline VLAN pairs	8.5(1)	8.5(1)	8.5(1)	8.5(1)	N/A	12.2(18)SXF4	12.2(18)SXF4	12.2(18)SXF4

1. Requires PFC2/3 or MSFC2/3.
2. Requires PFC2/3 or MSFC2/3.

Configuring the Catalyst 6500 Series Switch for Command and Control Access to IDSM-2

You must configure the Catalyst 6500 series switch to have command and control access to IDSM-2. This section describes how to configure the switch to have command and control access, and contains the following topics:

Catalyst Software

To configure the Catalyst 6500 series switch to have command and control access to IDSM-2, follow these steps:

Step 1 Log in to the console.

Step 2 Enter privileged mode:

```
console> enable
```

Step 3 Put the command and control port into the correct VLAN:

```
console> (enable) set vlan command_and_control_vlan_number
idsm2_slot_number/command_and_control_port_number
```

Example:

```
console> (enable) set vlan 147 6/2
VLAN 147 modified.
VLAN 146 modified.
VLAN  Mod/Ports
---- -----------------------
147   2/5,2/16-18
      6/2
```

The command and control port number is always 2.

Step 4 Session to IDSM-2 and ping a network IP address:

```
console> session slot_number
idsm-2# ping network_ip_address
```

Example:

```
console> (enable) session 6
Trying IDS-6...
Connected to IDS-6.
Escape character is '^]'.

login: cisco
Password:
Last login: Thu Mar 3 09:40:53 from 127.0.0.11
***NOTICE***
This product contains cryptographic features and is subject to United States and local
country laws governing import, export, transfer and use. Delivery of Cisco cryptographic
products does not imply third-party authority to import, export, distribute or use
encryption. Importers, exporters, distributors and users are responsible for compliance
```

```
with U.S. and local country laws. By using this product you agree to comply with
applicable laws and regulations. If you are unable to comply with U.S. and local laws,
return this product immediately.

A summary of U.S. laws governing Cisco cryptographic products may be found at:
http://www.cisco.com/wwl/export/crypto/tool/stqrg.html

If you require further assistance please contact us by sending email to
export@cisco.com.

***LICENSE NOTICE***
There is no license key installed on the system.
Please go to http://www.cisco.com/go/license to obtain a new license or install a license.
idsm-2# ping 10.89.149.126
PING 10.89.149.126 (10.89.149.126): 56 data bytes
64 bytes from 10.89.149.126: icmp_seq=0 ttl=255 time=0.3 ms
64 bytes from 10.89.149.126: icmp_seq=1 ttl=255 time=0.3 ms
64 bytes from 10.89.149.126: icmp_seq=2 ttl=255 time=0.3 ms
64 bytes from 10.89.149.126: icmp_seq=3 ttl=255 time=0.3 ms
--- 10.89.149.126 ping statistics ---
4 packets transmitted, 4 packets received, 0% packet loss
round-trip min/avg/max = 0.3/0.3/0.3 ms
idsm-2# exit
console> (enable)
```

Step 5 Initialize IDSM-2.

For the procedure, see Initializing the Sensor, page 3-2.

Step 6 Ping the default router of IDSM-2.

Step 7 Verify the management station can ping, SSH or Telnet, and web browse to IDSM-2.

Cisco IOS Software

To configure the Catalyst 6500 series switch to have command and control access to IDSM-2, follow these steps:

Step 1 Log in to the console.

Step 2 Enter global configuration mode:

```
router# configure terminal
```

Step 3 Put the command and control port into the correct VLAN:

```
router (config)# intrusion-detection module module_number management-port access-vlan
vlan_number
```

Example:

```
router (config)# intrusion-detection module 11 management-port access-vlan 146
```

Step 4 Verify that you have connectivity by sessioning in to IDSM-2 and pinging a network IP address:

```
router# session slot module_number processor 1
idsm-2# ping network_ip_address
```

Example:

```
router# session slot 11 processor 1
```

```
The default escape character is Ctrl-^, then x.
You can also type 'exit' at the remote prompt to end the session
Trying 127.0.0.91 ... Open

login: cisco
Password:
***NOTICE***
This product contains cryptographic features and is subject to United States and local
country laws governing import, export, transfer and use. Delivery of Cisco cryptographic
products does not imply third-party authority to import, export, distribute or use
encryption. Importers, exporters, distributors and users are responsible for compliance
with U.S. and local country laws. By using this product you agree to comply with
applicable laws and regulations. If you are unable to comply with U.S. and local laws,
return this product immediately.

A summary of U.S. laws governing Cisco cryptographic products may be found at:
http://www.cisco.com/wwl/export/crypto/tool/stqrg.html

If you require further assistance please contact us by sending email to
export@cisco.com.
***LICENSE NOTICE***
There is no license key installed on the system.
Please go to http://www.cisco.com/go/license
to obtain a new license or install a license.
idsm-2# ping 10.89.149.254
PING 10.89.149.254 (10.89.149.254): 56 data bytes
64 bytes from 10.89.149.254: icmp_seq=0 ttl=255 time=0.2 ms
64 bytes from 10.89.149.254: icmp_seq=1 ttl=255 time=0.2 ms
64 bytes from 10.89.149.254: icmp_seq=2 ttl=255 time=0.2 ms
64 bytes from 10.89.149.254: icmp_seq=3 ttl=255 time=0.2 ms
--- 10.89.149.254 ping statistics ---
4 packets transmitted, 4 packets received, 0% packet loss
round-trip min/avg/max = 0.2/0.2/0.2 ms
idsm-2# exit
[Connection to 127.0.0.91 closed by foreign host]
router#
```

Step 5 Initialize IDSM-2.

For the procedure, see Initializing the Sensor, page 3-2.

IDSM-2 Sensing Modes

IDSM-2 supports three sensing modes:

- Promiscuous mode—When IDSM-2 was introduced, promiscuous mode was the only sensing mode supported on IDSM-2 and it is the default sensing mode for both data ports. For more information on promiscuous mode on the sensor, see Promiscuous Mode, page 5-14.

  In promiscuous mode, IDSM-2 passively monitors network traffic copied to its data ports by the Catalyst switch. The data ports operate as 802.1q trunks and you can configure the two data ports to trunk the same or different VLANs. The Catalyst switch uses either SPAN or VACL capture to copy specific traffic to the data ports. You can send the same or different traffic to the two data ports. Because IDSM-2 is passive in this mode, it cannot drop packets to block a network intrusion attempt, but you can configure it to send TCP resets to both sides of the network connection to try to break the connection. For more information on TCP reset, see TCP Reset Interfaces, page 5-6. For the procedure for configuring SPAN, see Configuring SPAN, page 15-9. For the procedure for configuring VACL capture, see Configuring VACL Capture, page 15-13.

Note Because the Catalyst switch does not forward traffic received from a capture destination port, IDSM-2 cannot send TCP resets over the data ports to try to block an intrusion. Therefore, a separate reset port available only in promiscuous mode is reserved for this purpose.

- Inline mode—Beginning with IPS 5.0(1), you can configure IDSM-2 to be an active network device in inline interface pair mode. The two data ports operate together to bridge two VLANs through IDSM-2. You configure each data port as an access port and assign a different VLAN to each data port. IDSM-2 bridges the two VLANs by forwarding traffic between the two data ports. It inspects the traffic it receives on each data port and can either forward the packets to the other data port or drop the packet if it detects intrusion. For more information on inline mode for sensors, see Inline Interface Mode, page 5-15. You must configure the switch for inline mode (for more information see Configuring the Catalyst Series 6500 Switch for IDSM-2 in Inline Mode, page 15-19), and then create the inline interface pairs on IDSM-2 (for more information, see Configuring Inline Interface Pairs, page 5-15).
- Inline VLAN pair mode—Beginning with IPS 5.1(1), you can configure IDSM-2 in inline VLAN pair mode. IDSM-2 performs VLAN bridging between pairs of VLANs within the same data port operating as an 802.1q trunk. IDSM-2 inspects the traffic it receives on each VLAN in a VLAN pair and can either forward the packets on the other VLAN in the pair (on the same data port on which the packet was received) or drop the packet if an intrusion is detected. You can configure IDSM-2 to simultaneously bridge up to 255 VLAN pairs on each data port. IDSM-2 replaces the VLAN ID field in the 802.1q header of each packet with the ID of the VLAN on which the packet is forwarded. It drops any packets received on VLANs that are not assigned to an inline VLAN pair. For more information on inline VLAN pair mode, see Inline VLAN Pair Mode, page 5-17. For the procedure for configuring IDSM-2 inline VLAN pair mode, see Configuring the Catalyst Series 6500 Switch for IDSM-2 for Inline VLAN Pair Mode, page 15-21.

  Note You are responsible for coordinating the IPS and switch configuration to make sure each of the VLANS associated with an inline VLAN pair is also an allowed VLAN for the data port trunk.

  You can mix sensing modes on IDSM-2, for example, you can configure one data port for promiscuous mode and the other data port for inline VLAN pair mode. But because IDSM-2 only has two data ports and inline mode requires the use of both data ports as a pair, you cannot mix inline mode with either of the other two modes.

Configuring the Catalyst Series 6500 Switch for IDSM-2 in Promiscuous Mode

Traffic is captured for promiscuous analysis on IDSM-2 through SPAN or VACL capture (if you are running the Cisco IOS Firewall on the MSFC, you cannot use VACLs, but you can use the **mls ip ids** command). Port 1 (GigabitEthernet0/1) is used as the TCP reset port, port 2 (GigabitEthernet0/2) is the command and control port, and ports 7 and 8 (GigabitEthernet0/7 and GigabitEthernet0/8) are the monitoring ports. You can configure both monitoring ports to be either SPAN destination ports or VACL capture ports.

Caution If you configure both ports as monitoring ports, make sure that they are configured to monitor different traffic.

Caution You should not configure an IDSM-2 data port as both a SPAN destination port and a VACL capture port, because IDSM-2 will not receive traffic. This dual configuration (SPAN and VACL) causes problems on the switch and traffic is not sent properly.

Note Before Catalyst Software 8.4(3), IDSM-2 data ports defaulted to trunking all VLANs. In Catalyst Software 8.4(3) and later, IDSM-2 data ports default to trunking no VLANs. Make sure that the IDSM-2 ports are trunking the proper VLANs, especially if you upgrading from pre-8.4(3) to 8.4(3) or later.

This section contains the following topics:

Using the TCP Reset Interface

The IDSM-2 has a TCP reset interface—port 1. The IDSM-2 has a specific TCP reset interface because it cannot send TCP resets on its sensing ports.

If you have reset problems with the IDSM-2, try the following:

- If the sensing ports are access ports (a single VLAN), you need to configure the reset port to be in the same VLAN.
- If the sensing ports are dot1q trunk ports (multi-VLAN), the sensing ports and reset port all must have the same native VLAN, and the reset port must trunk all the VLANs being trunked by both the sensing ports.

Configuring SPAN

IDSM-2 can analyze Ethernet VLAN traffic from Ethernet or Fast Ethernet SPAN source ports, or you can specify an Ethernet VLAN as the SPAN source.

This section describes how to configure SPAN, and contains the following topics:

Catalyst Software

Use the **set span** command in privileged mode to enable SPAN to IDSM-2.

Note IDSM-2 port numbers are 7 or 8 only.

The following options apply:

- **disable**—Disables port monitoring.
- *module/port*—Source module and port numbers.
- *vlan*—Source VLAN numbers.
- *module/port*—Destination module and port numbers.
- **both**—Both receiving and transmitting traffic.
- **filter**—Applies filter to VLAN.
- **inpkts**—Enables/disables destination port incoming packets.
- **learning**—Enables/disables MAC address learning.
- **multicast**—Enables/disables multicast traffic.
- **rx**—Receiving traffic.
- **session**— Session number for SPAN session.
- **tx** —Transmitting traffic.

To enable SPAN on IDSM-2, follow these steps:

Step 1 Log in to the console.

Step 2 Enter privileged mode:

```
console> enable
```

Step 3 Enable SPAN to IDSM-2:

- From a source port:

```
console> (enable) set span 3/3 13/7
Destination     : Port 13/7
Admin Source    : Port 3/3
Oper Source     : Port 3/3
Direction       : transmit/receive
Incoming Packets: disabled
Learning        : enabled
Multicast       : enabled
Filter          : -

Session Number  : 1

console> (enable)
```

Note Use the **filter** keyword to monitor traffic on specific VLANs on source trunk ports.

- From a VLAN:

```
console> (enable) set span 650 13/7 rx
```

```
Destination : Port 13/7
Admin Source : VLAN 650
Oper Source : Port 11/1,13/1
Direction : receive
Incoming Packets: disabled
Learning : enabled
Multicast : enabled
Filter : -

Session Number : 1

console> (enable)
```

Step 4 Show the SPAN sessions:

```
console> (enable) show span

Destination      : Port 13/7
Admin Source     : VLAN 650
Oper Source      : Port 11/1,13/1
Direction        : receive
Incoming Packets: disabled
Learning         : enabled
Multicast        : enabled
Filter           : -

Session Number  : 1

Total local span sessions:  1
console> (enable)
```

Step 5 To disable the SPAN session that is sending traffic to IDSM-2:

```
console> (enable) set span disable session 1
This command will disable your span session.
Do you want to continue (y/n) [n]? y
Disabled Port 13/7 to monitor receive traffic of VLAN 650
console> (enable)
```

Note For more information on SPAN, refer to the Cisco Catalyst 6500 series switches configuration guides found at this URL: http://www.cisco.com/en/US/products/hw/switches/ps708/products_installation_and_configuration_guides_list.html

Cisco IOS Software

Use the **monitor session** command in global configuration mode to enable SPAN on IDSM-2.

Note Use 1 or 2 for IDSM-2 data port numbers.

The following options apply:

- **interface**—SPAN source interface
- **remote**—SPAN source Remote

- **vlan**— SPAN source VLAN
- **GigabitEthernet**— GigabitEthernet IEEE 802.3z
- **Port-channel**— Ethernet Channel of interfaces
- **,**— Specify another range of interfaces
- **-**— Specify a range of interfaces
- **both**— Monitor received and transmitted traffic
- **rx**— Monitor received traffic only
- **tx**— Monitor transmitted traffic only
- **intrusion-detection-module**— SPAN destination intrusion detection module
- **destination**— SPAN destination interface or VLAN
- **filter**— SPAN filter VLAN
- **source**— SPAN source interface, VLAN
- **type**— Type of monitor session

To enable SPAN on IDSM-2, follow these steps:

Step 1 Log in to the console.

Step 2 Enter global configuration mode:

```
router# configure terminal
```

Step 3 Set the source interfaces for the monitor session:

```
router(config)# monitor session (session_number) source interface interface/port_number
[, | - | rx | tx | both]
```

Example:

```
router(config)# monitor session 1 source interface GigabitEthernet2/23 both
```

Step 4 Enable an IDSM-2 data port as a SPAN destination:

```
router(config)# monitor session (session_number) destination intrusion-detection-module
module_number data-port data_port_number
```

Example:

```
router(config)# monitor session 1 destination intrusion-detection-module 9 data-port 1
```

Step 5 Make sure autostate is included for the data port:

```
router(config)# intrusion-detection-module module_number data-port data_port_number
autostate include
```

Example:

```
router(config)# intrusion-detection-module 9 data-port 1 autostate include
```

This allows the switch virtual interface to stay up if the data port is the only port in the VLAN. The default is **no include**.

Step 6 (Optional) Enable PortFast for the data port:

```
router(config)# intrusion-detection-module module_number data-port data_port_number
portfast enable
```

Example:

```
router(config)# intrusion-detection-module 9 data-port 1 portfast enable
```

The default is disabled.

Step 7 (Optional) To disable the monitor session:

```
router(config)# no monitor session session_number
```

Step 8 (Optional) To filter the SPAN session so that only certain VLANs are seen from switch port trunks:

```
router(config)# monitor session (session_number) {filter vlan {vlan_ID} [, | - ]}
```

Example:

```
router(config)# monitor session 1 filter vlan 146
```

Step 9 Exit configuration mode:

```
router(config)# exit
```

Step 10 To show current monitor sessions:

```
router# show monitor session session_number
```

Example:

```
router# show monitor session 1
    Session 1
    ---------
    Type                   : Local Session
    Source Ports           :
        Both               : Gi2/23
    Destination Ports      : intrusion-detection-module 9 data-port 1
```

Note For more information on SPAN, refer to the Cisco Catalyst 6500 series switches configuration guides found at this URL: http://www.cisco.com/en/US/products/hw/switches/ps708/products_installation_and_configuration_guides_list.html

Configuring VACL Capture

You can set VACLs to capture traffic for IPS from a single VLAN or from multiple VLANs or from FLexWAN2 ports on the 7600 router when using Cisco IOS software.

This section describes how to configure VACL capture, and contains the following topics:

- Catalyst Software, page 15-14
- Cisco IOS Software, page 15-15

Catalyst Software

Note Port 1 is set as the TCP reset port. Ports 7 and 8 are the sensing ports and can be configured as security ACL capture ports. By default, in Catalyst Software 8.4(1) and earlier releases, ports 7 and 8 are configured as trunk ports and trunk all VLANs on which a security ACL has been applied with the capture feature. If you want to monitor traffic from specific VLANs only, you need to clear the VLANs that you do not want to monitor so that they are not trunked to ports 7 and 8.

Use the **set security acl** command to configure security ACL capture ports.

The following options apply:

- **ACL**—Sets security ACL features
  - **capture-port**—Sets ports for ACL capture
  - **cram**—Sets security ACL cram
  - **ip** —Sets IP security ACL features
  - **ipx**—Sets IPX security ACL features
  - **mac**—Sets MAC security ACL features
  - **map**— Sets security ACL to VLAN mapping
- **permit**—Specifies packets to forward
- **deny**—Specifies packets to reject
- **redirect**—Specifies packets to redirect to ports
- **before**—Inserts ACE before a specified ace in editbuffer
- **capture**—Makes a copy of this flow in capture ports
- **modify**— Modifies a specified ACE in editbuffer

To configure VACLs to capture IPS traffic on VLANs, follow these steps:

Step 1 Log in to the console.

Step 2 Enter privileged mode.

```
console> enable
```

Step 3 Create the VACL to capture traffic. Specify what traffic is permitted, denied, and captured:

```
console> (enable) set security acl ip acl_name permit ip [permit (...) | deny (...)]
capture
```

Note Only permitted traffic can be captured. If you want to permit traffic but not capture it, do no use the **capture** keyword

Example:

```
console> (enable) set security acl ip CAPTUREALL permit ip any any capture
CAPTUREALL editbuffer modified. Use 'commit' command to apply changes.
```

Step 4 Commit the VACL:

```
console> (enable) commit security acl CAPTUREALL
```

```
ACL commit in progress.
```

Committing the VACL writes the VACL and associated ACEs to NVRAM.

Step 5 Map the VACL to the VLANs:

```
console> (enable) set security acl map acl_name vlan_number
```

Example:

```
console> (enable) set security acl map CAPTUREALL 650
Mapping in progress.

ACL CAPTUREALL successfully mapped to VLAN 650.
```

Step 6 Configure IDSM-2 ports (port 7 or 8) to be capture ports:

```
console> (enable) set security acl capture module_number/port_number
```

Example:

```
console> (enable) set security acl capture 2/7
Successfully set 2/7 to capture ACL traffic.
```

Note For more information on trunk ports and ACLs, refer to the Cisco Catalyst 6500 series switches configuration guides found at this URL: http://www.cisco.com/en/US/products/hw/switches/ps708/products_installation_and_configuration_guides_list.html

Cisco IOS Software

Use the following commands to configure VACLs to capture IPS traffic on VLANs.

The following options apply:

- **ip-access-list**—Named access list
  - **extended**—Extended Access List
  - **hardware**—Enable Hardware Fragment Handling
  - **log-update**—Control access list log updates
  - **logging**—Control access list logging
  - **resequence**—Resequence Access List
  - **standard**—Standard Access List

To configure VACLs to capture IPS traffic on VLANs, follow these steps:

Step 1 Log in to the console.

Step 2 Enter global configuration mode:

```
router# configure terminal
```

Step 3 Define the ACL:

```
router(config)# ip access-list [standard | extended] acl_name
```

Example:

```
router(config)# ip access-list standard CAPTUREALL
router(config-std-nacl)# exit
```

Step 4 Define the VLAN access map:

```
router(config)# vlan access-map map_name [0-65535]
```

Step 5 Configure a match clause in a VLAN access map sequence:

```
router(config-access-map)# match [ip address {1-199 | 1300-2699 | acl_name}]
```

Step 6 Configure an action clause in the VLAN access map sequence to accompany the preceding match clause:

```
router(config-access-map)# action forward capture
```

Step 7 Apply the VLAN access-map to the specified VLANs:

```
router(config)# vlan filter map_name vlan-list vlan_list
```

Step 8 Configure the IDSM-2 data ports to capture the captured-flagged traffic:

```
router(config)# intrusion-detection module module_number data-port data_port_number
capture allowed-vlan capture_vlans
```

Note When the switch is routing traffic, you should configure IDSM-2 to monitor all VLANs being routed. If you apply the VACL to a FlexWan2 port, you need to configure IDSM-2 to monitor all VLANs.

Step 9 Enable the capture function on IDSM-2:

```
router(config)# intrusion-detection module module_number data-port data_port_number
capture
```

This example shows the output from the **show run** command:

```
router# show run
intrusion-detection module 4 data-port 1 capture allowed-vlan 450,1002-1005
intrusion-detection module 4 data-port 1 capture
.
.
.
vlan access-map CAPTUREALL 10
match ip address MATCHALL
action forward capture
.
.
vlan filter CAPTUREALL vlan-list 450,1002-1005
.
ip access-list extended MATCHALL
permit ip any any
router#
```

Step 10 Make sure autostate is included for the data port:

```
router(config)# intrusion-detection-module module_number data-port data_port_number
autostate include
```

Example:

```
router(config)# intrusion-detection-module 4 data-port 1 autostate include
```

This allows the switch virtual interface to stay up if the data port is the only port in the VLAN. The default is **no include**.

Step 11 (Optional) Enable PortFast for the data port:

```
router(config)# intrusion-detection-module module_number data-port data_port_number
portfast enable
```

Example:

```
router(config)# intrusion-detection-module 4 data-port 1 portfast enable
```

The default is disabled.

Note For more information on autostate and PortFast, refer to the Cisco Catalyst 6500 series switches configuration guides found at this URL: http://www.cisco.com/en/US/products/hw/switches/ps708/products_installation_and_configuration_guides_list.html

Configuring the mls ip ids Command

This section describes how to use the **mls ip ids** command to capture IPS traffic, and contains the following topics:

- Catalyst Software, page 15-17
- Cisco IOS Software, page 15-18

Catalyst Software

When you are running the Cisco IOS Firewall on the MSFC, you cannot use VACLs to capture traffic for IDSM-2, because you cannot apply VACLs to a VLAN in which you have applied an IP inspect rule for the Cisco IOS Firewall. However, you can use the **mls ip ids** command to designate which packets are captured. Packets that are permitted by the ACL are captured. Those denied by the ACL are not captured. The **permit/deny** parameter does not affect whether a packet is forwarded to destination ports. Packets coming into that router interface are checked against the IPS ACL to determine if they should be captured. The **mls ip ids** command is applied as part of the MSFC configuration instead of the supervisor configuration. The **mls ip ids** command only captures incoming traffic. Use the **mls ip ids** command on both the client-side router interface and server-side router interface, so that both directions of the connection are captured.

To use the **mls ip ids** command to capture IPS traffic, follow these steps:

Step 1 Log in to the MSFC.

Step 2 Enter privileged mode:

```
console> enable
```

Step 3 Enter configuration mode:

```
router# configure terminal
```

Step 4 Configure an ACL to designate which packets will be captured:

```
router(config)# ip access-list extended word
```

Step 5 Select the interface that carries the packets to be captured:

```
router(config)# interface interface_name
```

Step 6 Apply the ACL created in Step 4 to the interface selected in Step 5:

```
router(config-if)# mls ip ids word
```

Step 7 Log in to the supervisor engine.

Step 8 Enter privileged mode.

```
console> enable
```

Step 9 On the supervisor engine, add the IDSM-2 monitoring port (port 7 or 8) to the VACL capture list:

```
console> (enable) set security acl capture module_number/port_number
```

Caution For IDSM-2 to capture all packets marked by the **mls ip ids** command, port 7 or 8 of IDSM-2 must be a member of all VLANs to which those packets are routed.

Cisco IOS Software

When you are using ports as router interfaces rather than switch ports, there is no VLAN on which to apply a VACL.

You can use the **mls ip ids** command to designate which packets are captured. Packets that are permitted by the ACL are captured. Those denied by the ACL are not captured. The **permit/deny** parameter does not affect whether a packet is forwarded to destination ports. Packets coming into that router interface are checked against the IPS ACL to determine if they should be captured.

To use the **mls ip ids** command to capture IDS traffic, follow these steps:

Step 1 Log in to the console.

Step 2 Enter global configuration mode:

```
router# configure terminal
```

Step 3 Configure an ACL to designate which packets will be captured:

```
router(config)# ip access-list extended word
```

Step 4 Select the interface that carries the packets to be captured:

```
router(config)# interface interface_name
```

Step 5 Specify the capture VLANs:

```
router(config)# intrusion-detection module module_number data-port data_port_number
capture allowed-vlan capture_vlans
```

Example:

```
router(config)# intrusion-detection module 4 data-port 1 capture allowed-vlan 165
```

Step 6 Apply the ACL created in Step 4 to the interface selected in Step 5:

```
router(config-if)# mls ip ids word
```

Caution For IDSM-2 to capture all packets marked by the **mls ip ids** command, data port 1 or data port 2 of IDSM-2 must be a member of all VLANs to which those packets are routed.

Configuring the Catalyst Series 6500 Switch for IDSM-2 in Inline Mode

You can use IDM or the CLI to configure IDSM-2 to operate in inline mode between two separate VLANs (one VLAN for each side of IDSM-2). To prepare IDSM-2 for inline mode, you must configure the switch as well as IDSM-2. Configure the switch first, then configure the IDSM-2 interfaces for inline mode. For the procedure for configuring IDSM-2 to run in inline mode, see Configuring Inline Interface Pairs, page 5-15.

This section contains the following topics:

Catalyst Software

You configure IDSM-2 monitoring ports as trunk ports for inline operation for Catalyst software 8.4(1) or later with Supervisor Engine 1a, Supervisor Engine 2, Supervisor Engine 32, or Supervisor Engine 720. Because the native VLAN is the same as the sole VLAN being trunked, the traffic is not 802.1q encapsulated.

Caution Before Catalyst software 8.4.(3), the default configuration for IDSM-2 ports 7 and 8 is to trunk all VLANs 1 to 4094. If you clear the IDSM-2 configuration (**clear configuration** *module_number*), IDSM-2 trunks all VLANs. If the IDSM-2 interfaces are configured for inline, spanning tree loops will likely be created and a storm will occur. A storm is numerous packets looping and never reaching their destination.

To configure the monitoring ports on IDSM-2 for inline operation, follow these steps:

Step 1 Log in to the console.

Step 2 Enter privileged mode.

```
console> enable
```

Step 3 Set the native VLAN for each IDSM-2 monitoring port:

```
console (enable)> set vlan vlan_number slot_number/port_number
```

Example:

```
console (enable)> set vlan 651 9/7
console (enable)> set vlan 652 9/8
```

Step 4 Clear all VLANs from each IDSM-2 monitoring port except for the native VLAN on each port (651 for port 7 and 652 on port 8):

```
console (enable)> clear trunk slot_number/port_number vlan_range
```

Example:

```
console (enable)> clear trunk 9/7 1-650,652-4094
console (enable)> clear trunk 9/8 1-651,653-4094
```

Step 5 Use the IPS CLI or IDM to pair the interfaces from Step 3 on IDSM-2. For more information, see Configuring Inline Interface Pairs, page 5-15.

Cisco IOS Software

Configure the IDSM-2 monitoring ports as access ports for inline operation.

To configure inline VLANs, follow these steps:

Step 1 Log in to the console.

Step 2 Enter global configuration mode:

```
router# configure terminal
```

Step 3 Select the VLANs the IDSM-2 will link.

Step 4 Configure each IDSM-2 data port to be on a single VLAN.

```
router(config)# intrusion-detection module slot_number data-port {1 | 2} access-vlan
vlan_number
router(config)# exit
```

Example:

```
router(config)# intrusion-detection module 13 data-port 1 access-vlan 661
router(config)# intrusion-detection module 13 data-port 2 access-vlan 662
router(config)# exit
```

Step 5 Verify the configuration:

Note In these examples, the IDSM-2 in slot 13 is inline between VLANs 661 and 662. The IDSM-2 data port 1 is on VLAN 661 and data port 2 is on VLAN 662.

a. Verify the IDSM-2 intrusion detection settings:

```
router# show run | include intrusion-detection
intrusion-detection module 13 management-port access-vlan 147
intrusion-detection module 13 data-port 1 access-vlan 661
intrusion-detection module 13 data-port 2 access-vlan 662
router#
```

b. Verify that the IDSM-2 data port 1 is an access port on VLAN 661:

```
router# show intrusion-detection module slot_number data-port data_port_number state
```

Example:

```
router# show intrusion-detection module 13 data-port 1 state
Intrusion-detection module 13 data-port 1:

Switchport: Enabled
Administrative Mode: static access
Operational Mode: static access
Administrative Trunking Encapsulation: dot1q Operational Trunking Encapsulation:
native Negotiation of Trunking: Off Access Mode VLAN: 661 (inline-vlan-1) Trunking
Native Mode VLAN: 1 (default) Trunking VLANs Enabled: NONE Pruning VLANs Enabled:
2-1001 Vlans allowed on trunk:661 Vlans allowed and active in management domain: 661
Vlans in spanning tree forwarding state and not pruned: 661
Administrative Capture Mode: Disabled
Administrative Capture Allowed-vlans: <empty>
```

c. Verify the VLAN number:

```
router# show vlan id vlan-number
```

Example:

```
router# show vlan id 661
VLAN Name                             Status    Ports
---- -------------------------------- --------- -------------------------------
661  ward-attack3                     active    Gi3/2, Gi13/d1

VLAN Type  SAID       MTU   Parent RingNo BridgeNo Stp  BrdgMode Trans1 Trans2
---- ----- ---------- ----- ------ ------ -------- ---- -------- ------ ------
661  enet  100661     1500  -      -      -        -    -        0      0

Remote SPAN VLAN
----------------
Disabled

Primary Secondary Type              Ports
------- --------- ----------------- ------------------------------------------
router#
```

Step 6 Use the IPS CLI or IDM to pair the interfaces from Step 4 on IDSM-2. For more information, see Configuring Inline VLAN Pairs, page 5-18.

Configuring the Catalyst Series 6500 Switch for IDSM-2 for Inline VLAN Pair Mode

You can use IDM or the CLI to configure IDSM-2 to operate in inline VLAN pair mode. To prepare IDSM-2 for inline VLAN pair mode, you must configure the switch as well as IDSM-2. Configure the switch first, then configure the IDSM-2 interfaces for inline VLAN pair mode. For the procedure for configuring IDSM-2 to run in inline VLAN pair mode, see Configuring Inline VLAN Pairs, page 5-18.

This section contains the following topics:

Catalyst Software

You configure IDSM-2 monitoring ports as trunk ports for inline VLAN pair mode for Catalyst software 8.4(1) or later with Supervisor Engine 1a, Supervisor Engine 2, Supervisor Engine 32, or Supervisor Engine 720.

To configure the monitoring ports on IDSM-2 for inline vlan pair mode, follow these steps:

Step 1 Log in to the console.

Step 2 Enter privileged mode.

```
console> enable
```

Step 3 Clear all VLANs from the IDSM-2 monitoring port:

```
console (enable)> clear trunk slot_number/port_number 1-4094
```

Example:

```
console (enable)> clear trunk 9/7 1-4094
```

Note Before Catalyst software 8.4.(3), the value for the VLAN range when clearing VLANs from the IDSM-2 monitoring port was 1-1005, 1024-4094. In later versions you can clear the entire VLAN range, 1-4094.

Step 4 Configure the IDSM-2 monitoring port to trunk the VLANs to be paired:

```
console (enable)> set trunk slot_number/port_number vlans_to_be_paired
```

Example:

```
console (enable)> set trunk 9/7 651,652
```

Step 5 Set the native VLAN for the IDSM-2 monitoring port to a value other than the paired VLANs used in Step 4:

```
console (enable)> set vlan vlan-number slot_number/port_number
```

Example:

```
console (enable)> set vlan 1 9/7
```

The default native VLAN is VLAN 1.

Step 6 Repeat Step 4 for other VLANs to be paired on the IDSM-2 monitoring port.

Step 7 To configure the other monitoring port, repeat Steps 3 through 6.

Step 8 Use the IPS CLI or IDM to pair the VLANs from Step 4 on IDSM-2. For more information, see Configuring Inline VLAN Pairs, page 5-18.

Cisco IOS Software

Configure the IDSM-2 monitoring ports as trunk ports for inline VLAN pair operation.

To configure inline VLAN pairs, follow these steps:

Step 1 Log in to the console.

Step 2 Enter global configuration mode:

```
router# configure terminal
```

Step 3 Configure one IDSM-2 data port to trunk the VLANs to be paired:

```
router(config)# intrusion-detection module slot_number data-port data_port_number trunk
allowed-vlan vlans_to_be_paired
router(config)# exit
```

Example:

```
router(config)# intrusion-detection module 13 data-port 1 trunk allowed-vlan 661,662
router(config)# exit
```

Step 4 Verify the configuration:

Note In these examples, data port 1 of IDSM-2 in slot 13 is trunking VLANs 661 and 662.

a. Verify the IDSM-2 intrusion detection settings:

```
router# show run | include intrusion-detection
intrusion-detection module 13 management-port access-vlan 147
intrusion-detection module 13 data-port 1 trunk allowed-vlan 661,662
router#
```

b. Verify that the IDSM-2 data port is trunking the proper VLANs:

```
router# show intrusion-detection module slot_number data-port data_port_number state
```

Example:

```
router# show intrusion-detection module 13 data-port 1 state
Intrusion-detection module 13 data-port 1:

Switchport: Enabled
Administrative Mode: trunk
Operational Mode: trunk
Administrative Trunking Encapsulation: dot1q
Operational Trunking Encapsulation: dot1q
Negotiation of Trunking: Off
Access Mode VLAN: 1 (default)
Trunking Native Mode VLAN: 1 (default)
Trunking VLANs Enabled: 661,662
Pruning VLANs Enabled: 2-1001
Vlans allowed on trunk: 661-662
Vlans allowed and active in management domain: none
Vlans in spanning tree forwarding state and not pruned:
   none
Administrative Capture Mode: Disabled
```

```
Administrative Capture Allowed-vlans: empty
Autostate mode: excluded
Portfast mode: default

router#
```

Step 5 Use the IPS CLI or IDM to pair the VLANs from Step 3 on IDSM-2. For more information, see Configuring Inline VLAN Pairs, page 5-18.

Configuring EtherChannel Load Balancing

This section describes how to configure ECLB on IDSM-2. It contains the following topics:

Overview

Supervisor Engines in the Catalyst 6500 series chassis recognize IDSM-2 devices that are running IPS 5.*x* and greater as EtherChannel devices. This lets you install up to eight IDSM-2 devices in the same chassis.

The IDSM-2 in the Catalyst 6500 series switch has eight internal ports. Only four of these ports are used. Port 1 is a TCP/IP reset port. Port 2 is the command and control port. Ports 7 and 8 are the sensing ports for Catalyst software and data ports 1 and 2 for Cisco IOS software. The other ports are not used.

The backplane is 1000 Mbps, which is why IDSM-2 shows 1000 Mbps even though it can only handle about 600 Mbps of performance. ECLB allows up to eight IDSM-2 devices to participate in the load balancing on either port 7 or port 8.

EtherChannel and the Three Sensing Modes

EtherChannel provides load balancing and failover between multiple IDSM-2s in all three sensing modes. IDSM-2 does not participate in EtherChannel protocols, such as LACP or PAgP. Cisco IOS only allows load balancing using **src-dst-ip** algorithm so that all packets between a given pair of IP addresses are always mapped to the same channel. Catalyst software uses the **ip both** algorithm. This is necessary so IDSM-2 can correctly track the connections between two hosts.

Caution You cannot mix IDSM-2 data ports with other port types in an EtherChannel group. You must configure all data ports in an EtherChannel group identically.

EtherChannel and IDSM-2 operate in the following way in the three sensing modes:

- EtherChannel and promiscuous mode—When IDSM-2 operates in promiscuous mode, the two data ports operate independently of each other. If you configure the switch so that a data port has two or more IDSM-2s in a group, the switch distributes traffic between the IDSM-2s. This balances the traffic between multiple IDSM-2s. You should rebalance the channel when a data port goes to the `errDisabled` state, or IDSM-2 is shut down, powered down, or reset.
- EtherChannel and inline mode—When you configure multiple IDSM-2s for inline mode, you can load balance the traffic between the IDSM-2s by putting data port 1 of each IDSM-2 into one channel group and data port 2 of each IDSM-2 into another channel group.

Caution To make sure that the same traffic is assigned to the two data ports on each IDSM-2, you must assign the same EtherChannel index to both data ports on each of the IDSM-2s even though they are in different EtherChannel groups.

- EtherChannel and inline VLAN pair mode—When IDSM-2 is in inline on-a-stick mode, the two data ports operate independently of each other. The same restrictions apply as for promiscuous mode.

Enabling ECLB

This section describes how to enable ECLB for Cisco IOS and Catalyst software. It contains the following sections:

- Catalyst Software, page 15-25
- IOS Software, page 15-27

Catalyst Software

This section describes how to enable ECLB for the three sensing modes in Catalyst software. It contains the following topics:

- ECLB in Promiscuous and VLAN Pair mode, page 15-25
- ECLB in Inline Mode, page 15-26

ECLB in Promiscuous and VLAN Pair mode

For promiscuous mode and inline VLAN pair mode, add the single port (port 7 or port 8) from each IDSM-2 to an EtherChannel.

To configure the monitoring ports on IDSM-2 for ECLB in promiscuous or inline VLAN pair mode, follow these steps:

Step 1 Log in to the console.

Step 2 Enter privileged mode.

```
console> enable
```

Step 3 Configure each IDSM-2 for promiscuous or inline VLAN pair mode.

Use the IPS CLI or IDM to pair the VLANs on IDSM-2. For more information, see Configuring Inline VLAN Pairs, page 5-18.

Step 4 Add the IDSM-2 monitoring ports to an EtherChannel

```
console (enable)> set port channel slot_number/port_number channel_number
```

Example:

```
console (enable)> set port channel 1/7,7/7 1
```

Step 5 Set the distribution method:

```
console (enable)> set port channel all distribution ip both
Channel distribution is set to ip both.
console (enable)>
```

Step 6 Enable ECLB:

```
console (enable)> set port channel slot_number/port_number mode on
```

Example:

```
console (enable)> set port channel 1/7,7/7 mode on
```

ECLB in Inline Mode

For inline mode, add the single port 7 from each IDSM-2 to an EtherChannel and port 8 from each IDSM-2 to a different EtherChannel.

To configure the monitoring ports on IDSM-2 for ECLB in inline mode, follow these steps:

Step 1 Log in to the console.

Step 2 Enter privileged mode.

```
console> enable
```

Step 3 Configure each IDSM-2 for inline mode.

Use the IPS CLI or IDM to pair the interfaces on IDSM-2. For more information, see Configuring Inline Interface Pairs, page 5-15.

Step 4 Add the IDSM-2 monitoring port 7s to an EtherChannel:

```
console (enable)> set port channel slot_number/7 channel_A_number
```

Example:

```
console (enable)> set port channel 1/7,7/7 1
```

Step 5 Enable ECLB for that EtherChannel:

```
console (enable)> set port channel slot_number/7 mode on
```

Example:

```
console (enable)> set port channel 1/7,7/7 mode on
```

Step 6 Add the IDSM-2 monitoring port 8s to another EtherChannel:

```
console (enable)> set port channel slot_number/8 channel_B_number
```

Example:

```
console (enable)> set port channel 1/8,7/8 2
```

Step 7 Enable ECLB for that EtherChannel:

```
console (enable)> set port channel slot_number/8 mode on
```

Example:

```
console (enable)> set port channel 1/8,7/8 mode on
```

Step 8 Set the distribution method:

```
console (enable)> set port channel all distribution ip both
Channel distribution is set to ip both.
console (enable)>
```

IOS Software

Note IOS 12.2(18)SXF4 or later is required for inline mode.

This section describes how to enable ECLB for the three sensing modes in Cisco IOS software. It contains the following topics:

- Restoring Defaults, page 15-27
- ECLB in Promiscuous Mode, page 15-27
- ECLB in Inline Mode, page 15-30

Restoring Defaults

Use the **intrusion-detection module** *module_number* **data-port {1 | 2} default** command to restore the defaults to the specified data port. This command restores the following defaults: allowed VLANs, autostate, portfast, cost, and priority settings. If the data port belongs to a port channel, this command has no effect. This command is useful for clearing the data port before you add it to a port channel group.

This command is equivalent to using all of the following commands:

- **no intrusion-detection module** *module_number* **data-port {1 | 2} trunk allowed-vlan**
- **intrusion-detection module** *module_number* **data-port {1 | 2} access vlan**
- **intrusion-detection module** *module_number* **data-port {1 | 2} autostate include**
- **intrusion-detection module** *module_number* **data-port {1 | 2} portfast**
- **intrusion-detection module** *module_number* **data-port {1 | 2} spanning-tree cost**
- **intrusion-detection module** *module_number* **data-port {1 | 2} spanning-tree priority**

ECLB in Promiscuous Mode

Note For Cisco IOS version and supervisor requirements for EtherChannel load balancing on IDSM-2, see Table 15-1 on page 15-4.

Note Cisco IOS supports promiscuous IDSM-2 EtherChannel using VACL capture (not SPAN or monitor).

An EtherChannel balances the traffic load across the links in an EtherChannel by reducing part of the binary pattern formed from the addresses in the frame to a numerical value that selects one of the links in the channel.

EtherChannel load balancing can use MAC addresses, IP addresses, or Layer 4 port numbers, which can be source or destination or both source and destination addresses or ports. The selected mode applies to all EtherChannels configured on the switch. ECLB can also use MPLS Layer 2 information.

Use the option that provides the balance criteria with the greatest variety in your configuration. For example, if the traffic on an EtherChannel is going only to a single MAC address and you use the destination MAC address as the basis of ECLB, the EtherChannel always chooses the same link in the EtherChannel; using source addresses or IP addresses might result in better load balancing.

For more information on EtherChannel, refer to *Catalyst 6500 Series Cisco IOS Software Configuration Guide, 12.2SX*.

To configure ECLB for promiscuous operation on IDSM-2, follow these steps:

Step 1 Configure each IDSM-2 for promiscuous operation.

For the procedure, see Configuring Promiscuous Mode, page 5-15.

Note Make sure that all IDSM-2 VACL capture or SPAN or monitor configuration lines have been removed before configuring ECLB for IDSM-2.

Step 2 Log in to the console.

Step 3 Enter global configuration mode:

```
router# configure terminal
```

Step 4 Create the VACL:

```
router(config)# ip access-list extended vacl_name
```

Example:

```
router(config)# ip access-list extended idstest
```

Step 5 Add any access control entries, for example, `permit any any`:

```
router(config-ext-nacl)# permit ip any any
```

Step 6 Create at least one VLAN access map sequence:

```
router(config-ext-nacl)# vlan access-map vlan_access_map_name sequence_number
router(config-access-map)# match ip address vacl_name
router(config-access-map)# action forward capture
```

Example:

```
router(config)# vlan access-map idstestmap 10
router(config-access-map)# match ip address idstest
router(config-access-map)# action forward capture
```

Step 7 Apply the VLAN access map to the VLAN(s):

```
router(config-access-map)# vlan filter vlan_access_map_name vlan-list vlan-list
```

Example:

```
router(config)# vlan filter idstestmap vlan-list 50-60
```

Step 8 For each IDSM-2, add the desired data ports into the desired EtherChannel:

```
router(config)# intrusion-detection module module_number data-port data_port_number
channel-group channel_number
```

Example:

```
router(config)# intrusion-detection module 13 data-port 7 channel-group 3
router(config)# intrusion-detection module 12 data-port 7 channel-group 3
```

Each EtherChannel has a numbered port channel interface. You can configure a maximum of 64 port channel interfaces, numbered from 1 to 256.

Step 9 Configure ECLB:

```
router(config)# port-channel load-balance src-dst-ip
```

The default and only load balancing algorithm supported for IDSM-2 is **src-dst-ip**, which means EtherChannel uses the combination of source and destination IP addresses for its distribution method.

Step 10 Verify the load balancing:

```
router# show etherchannel load-balance
EtherChannel Load-Balancing Configuration:
        src-dst-ip

EtherChannel Load-Balancing Addresses Used Per-Protocol:
Non-IP: Source XOR Destination MAC address
  IPv4: Source XOR Destination IP address
  IPv6: Source XOR Destination IP address
  MPLS: Label or IP
```

Step 11 Set the VLANs to be captured to the EtherChannel:

```
router(config)# intrusion-detection port-channel channel_number capture allowed-vlan
vlan_list
```

Example:

```
router(config)# intrusion-detection port-channel 3 capture allowed-vlan 10
```

Step 12 Enable capture to the EtherChannel:

```
router(config)# intrusion-detection port-channel channel_number capture
```

Example:

```
router(config)# intrusion-detection port-channel 3 capture
```

Step 13 Make sure autostate is included for the channel group:

```
router(config)# intrusion-detection port-channel channel_number autostate include
```

Example:

```
router(config)# intrusion-detection port-channel 3 autostate include
```

This allows the switch virtual interface to stay up if the data port is the only port in the VLAN. The default is **no include**.

Step 14 (Optional) Enable PortFast for the channel group:

```
router(config)# intrusion-detection port-channel channel_number portfast enable
```

Example:

```
router(config)# intrusion-detection port-channel 3 portfast enable
```

The default is disabled.

Step 15 Exit global configuration mode:

```
router(config)# exit
```

Step 16 To save the changes:

```
router# write memory
```

Note For more information on autostate and PortFast, refer to the Cisco Catalyst 6500 series switches configuration guides found at this URL: http://www.cisco.com/en/US/products/hw/switches/ps708/products_installation_and_configuration_guides_list.html

ECLB in Inline Mode

Note Make sure that all IDSM-2 VACL capture or SPAN or monitor configuration lines have been removed before configuring ECLB for IDSM-2. You receive an error if you try to change the channel group to inline mode if you have capture enabled on any of the ports.

To configure ECLB for inline mode on IDSM-2, follow these steps:

Step 1 Log in to the console.

Step 2 Enter global configuration mode:

```
router# configure terminal
```

Step 3 For each IDSM-2, add all data port 1s into an EtherChannel:

```
router(config)# intrusion-detection module module_number data-port 1 port-channel
channel_number
```

Example:

```
router(config)# intrusion-detection module 1 data-port 1 port-channel 5
```

Each EtherChannel has a numbered port channel interface. You can configure a maximum of 64 port channel interfaces, numbered from 1 to 256. If the channel group and port channel have not been created, this command creates it with an empty allowed VLAN list. If the port channel exists, its allowed VLAN list, port fast, autostate, spanning tree cost, and priority settings are assigned to the data port.

Note You receive an error if you try to add a data port to a channel group that contains other port types or if you try to add another port type to a port channel containing one or more data ports.

Step 4 For each IDSM-2, add all data port 2s into a different EtherChannel:

```
router(config)# intrusion-detection module module_number data-port 2 port-channel
channel_number
```

Example:

```
router(config)# intrusion-detection module 1 data-port 2 port-channel 6
```

Step 5 Set the sensing mode to access (inline) and set the access VLAN for the channel group that contains the data port 1s:

```
router(config)# intrusion-detection port-channel channel_number access-vlan vlan_id
```

Example:

```
router(config)# intrusion-detection port-channel 5 access-vlan 1050
```

Note You receive an error message if the port channel does not exist or if the port channel is already configured for trunk or capture mode. You must create the port channel or remove the port channel from trunk or capture mode.

Step 6 Set the sensing mode to access (inline) and set the access VLAN for the channel group that contains the data port 2s:

```
router(config)# intrusion-detection port-channel channel_number access-vlan vlan_id
```

Example:

```
router(config)# intrusion-detection port-channel 6 access-vlan 10
```

Step 7 Configure ECLB:

```
router(config)# port-channel load-balance src-dst-ip
```

The default is **src-dst-ip**., which means EtherChannel uses the combination of source and destination IP addresses for its distribution method.

Example:

```
router(config)# port-channel load-balance src-dst-ip
```

Step 8 Verify ECLB:

```
router# show etherchannel load-balance
EtherChannel Load-Balancing Configuration:
        src-dst-ip

EtherChannel Load-Balancing Addresses Used Per-Protocol:
Non-IP: Source XOR Destination MAC address
  IPv4: Source XOR Destination IP address
  IPv6: Source XOR Destination IP address
  MPLS: Label or IP
```

Step 9 For access (inline) mode, set autostate to **include** each channel group:

```
router(config)# intrusion-detection port-channel channel_number autostate include
```

Example:

```
router(config)# intrusion-detection port-channel 5 autostate include
```

The default is **no include**. This prevents the switch virtual interface from going down if the data port is up and in the VLAN.

Step 10 (Optional) You can enable or disable PortFast for each channel group:

```
router(config)# intrusion-detection port-channel channel_number portfast enable
```

Example:

```
router(config)# intrusion-detection port-channel 5 portfast enable
```

The default is disabled.

Step 11 (Optional) Set the spanning tree path cost for each of the two channel groups:

```
router(config)# intrusion-detection port-channel channel_number spanning-tree cost
port_cost
```

Example:

```
router(config)# intrusion-detection port-channel 5 spanning-tree cost 4
```

Both channel groups must be set to the same port cost to make sure that data port 1 and data port 2 of each IDSM-2 are in the same state (forwarding versus blocking).

Step 12 (Optional) Set the spanning tree port priority for each of the two channel groups:

```
router(config)# intrusion-detection port-channel channel_number spanning-tree priority
priority
```

Example:

```
router(config)# intrusion-detection port-channel 5 spanning-tree priority 16
```

The possible port priority value is a multiple of 16 from 0 to 240. The default is 32.

Step 13 Exit global configuration mode:

```
router(config)# exit
```

Step 14 To save the changes:

```
router# write memory
```

Note For more information on autostate, PortFast, spanning tree, and priority, refer to the Cisco Catalyst 6500 series switches configuration guides found at this URL: http://www.cisco.com/en/US/products/hw/switches/ps708/products_installation_and_configuration_guides_list.html

ECLB in Inline VLAN Pair Mode

Note Make sure that all IDSM-2 VACL capture or SPAN or monitor configuration lines have been removed before configuring ECLB for IDSM-2. You receive an error if you try to change the channel group to inline VLAN pair mode if you have **capture** enabled on any of the ports.

To configure ECLB for inline VLAN pair mode on IDSM-2, follow these steps:

Step 1 Log in to the console.

Step 2 Enter global configuration mode:

```
router# configure terminal
```

Step 3 Add the data port (either data port 1 or data port 2) from each IDSM-2 to the Etherchannel:

```
router(config)# intrusion-detection module module_number data-port [1:2] port-channel
channel_number
```

Example:

```
router(config)# intrusion-detection module 1 data-port 1 port-channel 5
```

Each EtherChannel has a numbered port channel interface. You can configure a maximum of 64 port channel interfaces, numbered from 1 to 256. If the channel group and port channel have not been created, this command creates it with an empty allowed VLAN list. If the port channel exists, its allowed VLAN list, port fast, autostate, spanning tree cost, and priority settings are assigned to the data port.

Note You receive an error if you try to add a data port to a channel group that contains other port types or if you try to add another port type to a port channel containing one or more data ports.

Step 4 Set the sensing mode to trunk (inline VLAN pair) and set the allowed VLANs for the channel group that contains the data port 1s. Determine which VLANs are going to be paired (100 and 200, 101 and 201) and set the allowed VLAN list to include all VLANs in all the pairs:

```
router(config)# intrusion-detection port-channel channel_number trunk allowed-vlan
vlan_list
```

Example:

```
router(config)# intrusion-detection port-channel 5 trunk allowed-vlans 100,101,200,201
```

Note The allowed VLAN list on the switch must include all VLANs that are paired as inline VLAN pairs on IDSM-2. Otherwise, traffic may be dropped.

Note You receive an error message if the port channel does not exist or if the port channel is already configured for trunk or capture mode. You must create the port channel or remove the port channel from trunk or capture mode.

Step 5 Configure ECLB:

```
router(config)# port-channel load-balance src-dst-ip
```

The default is **src-dst-ip**., which means EtherChannel uses the combination of source and destination IP addresses for its distribution method.

Example:

```
router(config)# port-channel load-balance src-dst-ip
```

Step 6 Verify ECLB:

```
router# show etherchannel load-balance
EtherChannel Load-Balancing Configuration:
        src-dst-ip

EtherChannel Load-Balancing Addresses Used Per-Protocol:
Non-IP: Source XOR Destination MAC address
  IPv4: Source XOR Destination IP address
  IPv6: Source XOR Destination IP address
  MPLS: Label or IP
```

Step 7 For access (inline) mode, set autostate to **include** the channel group:

```
router(config)# intrusion-detection port-channel channel_number autostate include
```

Example:

```
router(config)# intrusion-detection port-channel 5 autostate include
```

The default is **no include**. This prevents the switch virtual interface from going down if the data port is up and in the VLAN.

Step 8 (Optional) You can enable or disable PortFast for the channel group:

```
router(config)# intrusion-detection port-channel channel_number portfast enable
```

Example:

```
router(config)# intrusion-detection port-channel 5 portfast enable
```

The default is disabled.

Step 9 (Optional) Set the spanning tree port cost for the channel group:

```
router(config)# intrusion-detection port-channel channel_number spanning-tree cost
port_cost
```

Example:

```
router(config)# intrusion-detection port-channel 5 spanning-tree cost 4
```

Step 10 (Optional) Set the spanning tree port priority for the channel group:

```
router(config)# intrusion-detection port-channel channel_number spanning-tree priority
priority
```

Example:

```
router(config)# intrusion-detection port-channel 5 spanning-tree priority 16
```

The possible port priority value is a multiple of 16 from 0 to 240. The default is 32.

Step 11 Exit global configuration mode:

```
router(config)# exit
```

Step 12 To save the changes:

```
router# write memory
```

Step 13 Use the IPS CLI or IDM to pair the VLANs from Step 4 on IDSM-2. For more information, see Configuring Inline VLAN Pairs, page 5-18.

Note For more information on autostate, PortFast, spanning tree, and priority, refer to the Cisco Catalyst 6500 series switches configuration guides found at this URL: http://www.cisco.com/en/US/products/hw/switches/ps708/products_installation_and_configuration_guides_list.html

Disabling ECLB

This section explains how to disable ECLB, and contains the following topics:

Catalyst Software

To disable ECLB, follow these steps:

Step 1 Log in to the console.

Step 2 Enter privileged mode.

```
console> enable
```

Step 3 Disable ECLB for promiscuous or inline VLAN pair mode:

```
console (enable)> set port channel slot_number/port_number mode off

Example:
console (enable)> set port channel 1/7,7/7 mode off
```

Step 4 Disable ECLB for inline mode:

a. Disable ECLB for one EtherChannel:

```
console (enable)> set port channel slot_number/7 mode off

Example:
console (enable)> set port channel 1/7,7/7 mode off
```

b. Disable ECLB for the other EtherChannel:

```
console (enable)> set port channel slot_number/8 mode off

Example:
console (enable)> set port channel 1/8,7/8 mode off
```

Cisco IOS Software

To disable ECLB for IDSM-2, follow these steps:

Step 1 Log in to the console.

Step 2 Enter global configuration mode:

```
router# configure terminal
```

Step 3 To remove a single IDSM-2 from the EtherChannel:

```
router(config)# no intrusion-detection module module_number data-port data_port_number
channel-group channel_number
```

Example:

```
router(config)# no intrusion-detection module 1 data-port 1 channel-group 5
```

Step 4 To remove the whole EtherChannel:

```
router(config)# no intrusion-detection module port-channel channel_number
```

Example:

```
router(config)# no intrusion-detection module port-channel 5
```

Note The VACL capture commands for IDSM-2 are left.

Verifying ECLB

This section explains how to verify your ECLB configuration, and contains the following topics:

- Catalyst Software, page 15-36
- Cisco IOS Software, page 15-37

Catalyst Software

To verify the ECLB configuration, follow these steps:

Step 1 Log in to the console.

Step 2 Enter privileged mode.

```
console> enable
```

Step 3 To see all EtherChannels:

```
console (enable)> show channel slot_number/port_number mode off
```

Example:

```
console> (enable) show channel
Channel Id   Ports
-----------  ----------------------------------------------
1669         1/7,7/7
1698         2/1-6
console> (enable)
```

Note In this output, an EtherChannel with ID 1669 is created to have two IDSM-2 data ports. Port 1/7 is for port 7 on the IDSM-2 in slot 1 while port 7/7 is for port 7 on the IDSM-2 in slot 7. Both IDSM-2s are configured for promiscuous operation. The switch load balances between each of the two IDSM-2 ports (one port on each IDSM-2).

Step 4 To see specific EtherChannel status:

```
console (enable)> show channel hash channel_id source_ip_addr dest_ip_addr
```

Example:

```
console> (enable) show channel hash 1669 10.20.2.1 10.20.5.3
Selected channel port: 1/7
console> (enable)
```

Note This output shows that traffic from 10.20.2.1 to 10.20.5.3 will be sent to port 1/7 (port 7 for the IDSM-2 in slot 1).

Cisco IOS Software

To verify the IDSM-2 ECLBconfiguration, follow these steps:

Step 1 Log in to the console.

Step 2 To see all EtherChannels:

```
router# show etherchannel
                Channel-group listing:
                -----------------------

Group: 10
----------
Group state = L2
Ports: 0   Maxports = 8
Port-channels: 1 Max Port-channels = 1
Protocol:    -

router#
```

Step 3 To see specific EtherChannel status:

```
router# show etherchannel 1 [summary | detail | port | port-channel | protocol]
```

Example:

```
router# show etherchannel 1 summary
Flags:  D - down        P - in port-channel
        I - stand-alone s - suspended
        H - Hot-standby (LACP only)
        R - Layer3      S - Layer2
        U - in use      f - failed to allocate aggregator

        u - unsuitable for bundling
Number of channel-groups in use: 1
Number of aggregators:           1

Group  Port-channel  Protocol    Ports
------+-------------+-----------+---------------------------

router#
```

Step 4 To see the ECLB setting:

```
router# show etherchannel load-balance
EtherChannel Load-Balancing Configuration:
        src-dst-ip
        mpls label-ip
```

```
EtherChannel Load-Balancing Addresses Used Per-Protocol:
Non-IP: Source XOR Destination MAC address
  IPv4: Source XOR Destination IP address
  IPv6: Source XOR Destination IP address
  MPLS: Label or IP
router#
```

Step 5 To see IDSM-2 data port information:

```
router# show intrusion-detection module module_number data-port data_port_number state

Example:
router# show intrusion-detection module 11 data-port 2 state
Intrusion-detection module 11 data-port 2:

Switchport: Enabled
Administrative Mode: trunk
Operational Mode: trunk
Administrative Trunking Encapsulation: dot1q
Operational Trunking Encapsulation: dot1q
Negotiation of Trunking: Off
Access Mode VLAN: 662 (ward-victim3)
Trunking Native Mode VLAN: 1 (default)
Trunking VLANs Enabled: NONE
Pruning VLANs Enabled: 2-1001
Vlans allowed on trunk:none
Vlans allowed and active in management domain: none
Vlans in spanning tree forwarding state and not pruned:
   none
Administrative Capture Mode: Disabled
Administrative Capture Allowed-vlans: empty
```

Administrative Tasks for IDSM-2

This section contains procedures that help you with administrative tasks for IDSM-2. It contains the following topics:

- Enabling Full Memory Tests, page 15-38
- Resetting IDSM-2, page 15-40

Enabling Full Memory Tests

When IDSM-2 initially boots, by default it runs a partial memory test. You can enable a full memory test in Catalyst software and Cisco IOS software.

This section describes how to enable full memory tests, and contains the following topics:

- Catalyst Software, page 15-39
- Cisco IOS Software, page 15-39

Catalyst Software

Use the **set boot device** *boot_sequence module_number* **mem-test-full** command to enable a full memory test. The full memory test takes about 12 minutes.

To enable a full memory test, follow these steps:

Step 1 Log in to the console.

Step 2 Enter privileged mode:

```
console> enable
```

Step 3 Enable the full memory test:

```
console> (enable) set boot dev cf:1 3 mem-test-full
Device BOOT variable = cf:1
Memory-test set to FULL
Warning: Device list is not verified but still set in the boot string.
console> (enable) set boot dev hdd:1 3 mem-test-full
Device BOOT variable = hdd:1
Memory-test set to FULL
Warning: Device list is not verified but still set in the boot string.
console> (enable)
```

The **set boot device** command can either contain **cf:1** or **hdd:1**.

Step 4 Reset IDSM-2.

For the procedure, see Resetting IDSM-2, page 15-40.

The full memory test runs.

Note A full memory test takes more time to complete than a partial memory test.

Cisco IOS Software

Use the **hw-module module** *module_number* **reset mem-test-full** command to enable a full memory test. The full memory test takes about 12 minutes.

To enable a full memory test, follow these steps:

Step 1 Log in to the console.

Step 2 Enable the full memory test:

```
router# hw-module module 9 reset mem-test-full
    Device BOOT variable for reset = <empty>
    Warning: Device list is not verified.

    Proceed with reload of module?[confirm]
    % reset issued for module 9
router#
```

Step 3 Reset IDSM-2.

For the procedure, see Resetting IDSM-2, page 15-40.

The full memory test runs.

Note A full memory test takes more time to complete than a partial memory test.

Resetting IDSM-2

If for some reason you cannot communicate with IDSM-2 through SSH, Telnet, or the switch **session** command, you must reset IDSM-2 from the switch console. The reset process requires several minutes.

This section contains the following topics:

Catalyst Software

To reset IDSM-2 from the CLI, follow these steps:

Step 1 Log in to the console.

Step 2 Enter privileged mode:

```
console> enable
```

Step 3 Reset IDSM-2 to the application partition or the maintenance partition:

```
console> (enable) reset module_number [hdd:1 | cf:1]
```

Note If you do not specify either the application partition (hdd:1 the default) or the maintenance partition (cf:1), IDSM-2 uses the boot device variable.

Example:

```
console> (enable) reset 3
2003 Feb 01 00:18:23 %SYS-5-MOD_RESET: Module 3 reset from console//
Resetting module 3... This may take several minutes.
2003 Feb 01 00:20:03 %SYS-5-MOD_OK: Module 3 is online.
console> (enable)
```

Caution If IDSM-2 is removed from the switch chassis without first being shut down, or the chassis loses power, you may need to reset IDSM-2 more than once. If IDSM-2 fails to respond after three reset attempts, boot the maintenance partition, and perform the instructions for restoring the application partition. For the procedure, see Installing the IDSM-2 System Image, page 17-27.

Cisco IOS Software

Use the **hw-module module** *slot_number* **reset** [**hdd:1** | **cf:1**] command in EXEC mode to reset IDSM-2. The reset process takes several minutes. IDSM-2 boots into the boot partition you specify. If you do not specify the boot string, the default boot string is used.

To reset IDSM-2 from the CLI, follow these steps:

Step 1 Log in to the console.

Step 2 Reset IDSM-2:

```
router# hw-module module module-number reset [hdd:1 | cf:1]
```

Note If you do not specify either the application partition (**hdd:1** the default) or the maintenance partition (**cf:1**), IDSM-2 uses the boot device variable.

Example:

```
router# hw-module module 8 reset
Device BOOT variable for reset =
Warning: Device list is not verified.
Proceed with reload of module? [confirm]
% reset issued for module 8
router#
```

Catalyst and Cisco IOS Software Commands

This section lists the Catalyst and Cisco IOS software commands that pertain to IDSM-2.

Note For more detailed information on Catalyst and Cisco IOS software commands, refer to the command references found on Cisco.com. For instructions on how to locate these documents, refer to *Documentation Roadmap for Cisco Intrusion Prevention System 5.1*.

This section contains the following topics:

Catalyst Software

This section lists supported and unsupported Catalyst Software Commands. It contains the following topics:

Supported Supervisor Engine Commands

IDSM-2 also supports the following supervisor engine CLI commands, which are described in more detail in the Catalyst 6500 Series Command References.

- **clear config** *module_number*

  Clears the configuration on the supervisor engine that is associated with the specified IDSM-2.
- **clear log** *module_number*

  Deletes all entries in the error log for the specified IDSM-2.
- **session** *slot_number*

  Logs in to the console of IDSM-2 from the switch console.
- **set module** commands (all other **set module** commands return an error message):
  - **set module name** *module_number*

    Sets the name of the module.
  - **set module power** *module_number* [**up | down]**

    Enables or disables power to the specified IDSM-2.
- **set port name** *module_number*

  Configures the name for the specified IDSM-2 port.
- **set span**

  Configures port 1 as a SPAN destination port. You cannot use port 1 on IDSM-2 as a SPAN source port.
- **set trunk**

  Configures trunk ports.
- **set vlan**

  Configures VLAN capture ports.
- **show config**

  Displays the supervisor engine NVRAM configurations.
- **show log**

  Displays the error logs for the specified IDSM-2.
- **show mac** *module_number*

  Displays the MAC counters for the specified IDSM-2.
- **show module** *module_number*

  With an IDSM-2 installed, displays `Intrusion Detection System Module` under `Module-Type`.
- **show port** *module_number*

  Displays the port status for the specified IDSM-2.
- **show port capabilities** [*module* | *module_number*]

  Displays the capabilities of the module and ports.
- **show test**

  Displays the errors reported from the diagnostic tests for both the SPAN port (port 1) and the management port (port 2) and the BIOS and CMOS boot results.

Unsupported Supervisor Engine Commands

The following supervisor engine CLI commands are not supported by IDSM-2:

- **set module** [**enable** | **disable**] *module_number*
- **set port broadcast**
- **set port channel**
- **set port cops**
- **set port disable**
- **set port enable**
- **set port flowcontrol**
- **set port gmrp**
- **set port gvrp**
- **set port host**
- **set port inlinepower**
- **set port jumbo**
- **set port membership**
- **set port negotiation**
- **set port protocol**
- **set port qos**
- **set port rsvp**
- **set port security**
- **set port speed**
- **set port trap**
- **set protocolfilter**
- **set rgmp**
- **set snmp**
- **set spantree**
- **set udld**
- **set vtp**

Cisco IOS Software

This section lists the Cisco IOS software commands that IDSM-2 supports. These commands are grouped according to mode.

This section contains the following topics:

EXEC Commands

The following commands are all performed in EXEC mode:

- **clock read-calendar**

  Updates the clock time to the calendar time.

- **clock set** *time date*

  Sets the current time and date.

- **clock update-calendar**

  Updates the calendar time to the clock time.

- **hw-module module** *module_number* **reset** [**cf:1** | **hdd:1**]

  Resets IDSM-2 into the partition specified by the boot device variable; if the boot device variable has not been set, IDSM-2 is reset to the application partition by default. Use the command **show boot device module** *module_number* to view the current setting of the boot device variable. **cf:1** is the maintenance partition. **hdd:1** is the application partition.

- **hw-module module** *module_number* **shutdown**

  Shuts down IDSM-2 so that it can be safely removed from the chassis.

- **reload**

  Reloads the entire switch.

- **session slot** *module_number* **processor** *processor_number*

  Logs in to the console of IDSM-2 from the switch console.

- **show boot device module** *module_number*

  Displays the current boot string for the specified module.

- **show diagnostic result module** *module_number*

  Displays the results of the online diagnostics that were performed when IDSM-2 was last booted up.

- **show interface port-channel** *channel_number*

  Displays the status of the port channel.

- **show intrusion-detection module** *module_number* **data-port {1 | 2} {state | traffic}**

  Displays the state or traffic statistics of the specified IDSM-2 data port.

- **show intrusion-detection module** *module_number* **management-port {state | traffic}**

  Displays the state or traffic statistics of the IDSM-2 management port.

- **show ip access-lists**

  Displays the current access lists.

- **show module [***module_number* **| all | version]**

  Displays the installed modules, versions, and states.

- **show monitor session** *session_number*

  Displays the SPAN source and destination for the specified session.

- **show running-config**

  Displays the configuration that is currently running.

- **show spanning-tree active**

  Displays spanning tree state information for active interfaces only.
- **show spanning-tree detail**

  Displays detailed spanning tree state information.
- **show spanning-tree summary [totals]**

  Displays the high level state of spanning tree. Does not show interface specific information.
- **show spanning-tree vlan** *vlan_number*

  Displays spanning tree state information for the specified VLAN. Includes list of ports on which those VLANs are forwarded or blocked.
- **show startup-config**

  Displays the saved configuration.
- **show vlan access-map**

  Displays all current VLAN access maps.

Configuration Commands

The following configuration commands are all performed in either global configuration mode, interface configuration mode, or VACL configuration submode:

- Global configuration mode
  - **boot device module** *number_number* {**cf:1** | **hdd:1**}

    Sets the default boot device for the specified module. **cf:1** boots to the MP and **hdd:1** boots to the AP. The **no** option clears the boot string, which sets the default boot device to the AP.
  - **clock calendar valid**

    Sets the current calendar time as the switch time on bootup.
  - **clock summer-time** *zone* **recurring**

    Sets the switch to use the summertime settings.
  - **clock timezone** *zone offset*

    Sets the timezone for the switch/IDSM-2.
  - **fabric switching-mode force busmode**

    Lets service modules that do not support packet recirculation, be forced into communicating through the chassis shared bus instead of the switched fabric. This forces the supervisor to handle the packet recirculation centrally and lets the service module communicate properly on VLANs meeting the conditions stated above. Other fabric enabled modules that are not affected by this problem continue to communicate through the switch fabric even if this command is enabled.
  - **[no] intrusion-detection module** *module_number* **data-port {1 | 2} access vlan** *vlan_id*

    Sets the data port to access (inline) mode and sets the access VLAN for the data port for the specified module.

– **[no] intrusion-detection module** *module_number* **data-port {1 | 2} autostate include**

  Includes (or excludes) the specified data port in the autostate calculation. When included, the switch virtual interface associated with an MSFC or WLAN port remains up while the module's data port is enabled. When excluded, the switch virtual interface associated with the MSFC or WAN port goes down if the specified module's data port is the only active port in the VLAN. The default is **no include**.

– **[no] intrusion-detection module** *module_number* **data-port {1 | 2} capture**

  Configures the specified data port as a capture destination port. You must also set the allowed VLAN list through the **intrusion-detection module** *module_number* **data-port {1 | 2} capture** command before any packets are captured. The IDSM-2 must be in promiscuous mode.

– **[no] intrusion-detection module** *module_number* **data-port {1 | 2} capture allowed-vlan** *vlan_list*

  Sets the allowed VLANs on the specified data port for packet capture. You must also enable capture mode on the data port through the **intrusion-detection module** *module_number* **data-port {1 | 2} capture** command before traffic is captured on the data port.

– **intrusion-detection module** *module_number* **data-port {1 | 2} default**

  Restores the allowed VLANs, autostate, PortFast, port cost, and priority settings for the specified data port to the default values. This command is useful to remove any configuration from a data port before you add it to a channel group.

– **[no] intrusion-detection module** *module_number* **data-port {1 | 2}port-channel** *channel_number*

  Adds the data port for the specified module to the channel group, which creates a port channel with the same numeric ID. If the channel group and port channel have not been created, this command creates it with an empty allowed VLAN list. The **no** option removes the data port from the channel group, restores the data port settings to their defaults, and deletes the port channel if it is empty.

– **[no] intrusion-detection module** *module_number* **data-port {1 | 2}portfast enable | disable [trunk]**

  Enables or disables PortFast on the data port. When PortFast is enabled, traffic is forwarded by the switch to the IDSM-2 data port while the spanning tree is being built. When disabled, traffic is inhibited until after the tree is built and the backplane port is in the forwarding state. The default is disabled. The trunk option enables or disables PortFast with the data port is configured as a trunk (in promiscuous or inline VLAN pairs mode).

– **[no] intrusion-detection module** *module_number* **data-port {1 | 2} spanning-tree cost** *path_cost*

  Sets the spanning tree path cost for the data port on the specified module. The **no** option restores the spanning tree cost for the data port on the specified module to the default cost value.

– **[no] intrusion-detection module** *module_number* **data-port {1 | 2} trunk allowed-vlan** *vlan_list*

  Sets the data port to trunking mode and sets the list of allowed VLANs on the data port for the specified module. The **no** option removes the data port from trunking mode and clears the list of allowed VLANs on the data port for the specified module.

– **intrusion-detection module** *module_number* **management-port access-vlan** *vlan_number*

  Sets the access VLAN for the IDSM-2 command and control port.

– **intrusion-detection module** *module_number* **data-port** *data_port_number* **capture allowed-vlan** *allowed_capture_vlan(s)*

  Configures the VLAN(s) for VACL capture.

– **intrusion-detection module** *module_number* **data-port** *data_port_number* **capture**

  Enables VACL capture for the specified IDSM-2 data port.

– [**no**] **intrusion-detection port-channel** *channel_number* **access vlan** *vlan_id*

  Sets all data ports in the specified port channel to access mode and sets the access VLAN for the data ports. The **no** option clears the list of allowed VLANs on the data ports of all modules in the specified port channel.

– **[no] intrusion-detection port-channel** *channel_number* **autostate include**

  Includes or excludes all data ports in the specified port channel from the autostate calculation. When included, the virtual switch interface associated with an MSFC or WLAN port remains up while the data port is enabled. When excluded, the virtual switch interface associated with the MSFC or WAN port goes down if the data port is the only active port in the VLAN. The data ports are excluded from the autostate calculations by default.

– **[no] intrusion-detection port-channel** *channel_number* **capture**

  Configures all data ports in the channel group as capture ports. The **no** option disables the capture function on all data ports in the channel group.

– [**no**] **intrusion-detection port-channel** *channel_number* **capture allowed-vlan** *vlan_id*

  Sets the list of capture VLANs on the data ports of all modules in the specified port channel. This command does not set the channel group to capture mode. Use the **intrusion-detection port-channel** *channel_number* **capture** command to set the channel group to capture mode. The **no** option clears the list of capture VLANs on the data ports of all modules in the specified port channel.

– **[no] intrusion-detection port-channel** *channel_number* **portfast enable | disable [trunk]**

  Enables or disables PortFast on the data ports in the port channel. When PortFast is enabled, traffic is forwarded by the switch to the data port while the spanning tree is being built. When disabled, traffic is inhibited until after the tree is built and the backplane port is in the forwarding state. Use the **trunk** option to enable or disable PortFast when the data port is configured as a trunk (in promiscuous or inline VLAN pair mode). Do not use the **trunk** option when the data ports are configured as access ports (inline mode). PortFast and PortFast trunk are disabled by default.

– [**no**] **intrusion-detection port-channel** *channel_number* **spanning-tree cost** *port_cost*

  Sets the spanning tree port cost for the data port on the specified module. The **no** option restores the spanning tree port cost for the data port on the specified module to the default value.

– [**no**] **intrusion-detection port-channel** *channel_number* **spanning-tree priority** *priority*

  Sets the spanning tree port priority for the data port on the specified module. The **no** option restores the spanning tree port priority for the data port on the specified module to the default value.

– [**no**] **intrusion-detection port-channel** *channel_number* **trunk allowed-vlan** *vlan_id*

  Sets the list of allowed VLANs on the data ports of all modules in the specified port channel. The **no** option clears the list of allowed VLANs on the data ports of all modules in the specified port channel.

- **ip access-list extended** *word*

  Creates access lists for use in the VACL maps.

- **[no] monitor session** *session_number* **destination intrusion-detection module** *module_number* **data-port {1 | 2}**

  Configures a SPAN destination port, which can be either a standard line card port or an IDSM-2 data port.

- **[no] monitor session** *session_number* {**source** {**interface** *interface_number*} | {**vlan** *vlan_id*}} [, | - | **rx** | **tx** | **both**]

  Sets the sources for a SPAN session.

- **[no] power enable module** *module_number*

  Powers IDSM-2 off or on.

- **[no] spanning tree mode {pvst | mst | rapid-pvst}**

  Selects the spanning tree protocol (PVST+, MST, or Rapid-PVST+) to be used globally on the switch. The default is PVST. MST is not supported for IDSM-2. The **no** option restores the spanning tree mode to the default.

- **vlan access-map** *map_name_sequence*

  Creates the VACL maps.

- **vlan filter** *map_name* **vlan-list** *vlans*

  Maps the VACL maps to VLANs.

• Interface configuration mode

- **switchport**

  Sets the interface as a switch port.

- **switchport access vlan** *vlan*

  Sets the access VLAN for the interface.

- **switchport capture**

  Sets the interface as a capture port.

- **switchport mode access**

  Sets the interface as an access port.

- **switchport mode trunk**

  Sets the interface as a trunk port.

- **switchport trunk allowed vlan** *vlans*

  Sets the allowed VLANs for trunk.

- **switchport trunk encapsulation dot1q**

  Sets dot1q as the encapsulation type.

- **switchport trunk native vlan** *vlan*

  Sets the native VLAN for the trunk port.

- VACL configuration submode
  - **action forward capture**

    Designates that matched packets should be captured.
  - **match ip address** [*1-199* | *1300-2699* | *acl_name*]

    Specifies filtering in the VACL.

CHAPTER 16

Configuring NM-CIDS

Note NM-CIDS does not operate in inline mode, only in promiscuous mode, therefore you cannot configure intrusion prevention.

Note NM-CIDS does not support bridged interfaces. Although Cisco IOS allows you to configure IDS using NM-CIDS on a bridged interface, NM-CIDS is not designed to inspect traffic on bridged interfaces, and the configuration does not work.

This chapter describes the tasks you need to perform to set up NM-CIDS and get it ready to receive traffic. After that you are ready to configure intrusion detection. This chapter contains the following sections:

Configuration Sequence

Perform the following tasks to configure NM-CIDS:

1. Configure the IDS interfaces on the router.

   For the procedure, see Configuring IDS-Sensor Interfaces on the Router, page 16-2.
2. Log in to NM-CIDS.

   For the procedure, see Establishing NM-CIDS Sessions, page 16-3.
3. Initialize NM-CIDS.

   Run the **setup** command to initialize NM-CIDS.

   For the procedure, see Initializing the Sensor, page 3-2.
4. Configure NM-CIDS to capture traffic for intrusion detection analysis.

For the procedure, see Configuring Packet Capture, page 16-5.

5. Create the service account.

   For the procedure, see Creating the Service Account, page 4-13.

6. Perform the other initial tasks, such as adding users, trusted hosts, and so forth.

   For the procedures, see Chapter 4, "Initial Configuration Tasks."

7. Configure intrusion detection.

   For the procedures, see Chapter 16, "Configuring NM-CIDS," Chapter 7, "Defining Signatures," and Chapter 10, "Configuring Attack Response Controller for Blocking and Rate Limiting.".

8. Perform administrative tasks to keep your NM-CIDS running smoothly.

   For the procedures, see Chapter 13, "Administrative Tasks for the Sensor," and Administrative Tasks, page 16-6.

9. Upgrade the IPS software with new signature updates and service packs.

   For more information, see Obtaining Cisco IPS Software, page 18-1.

10. Reimage the boot helper and bootloader when needed.

    For the procedures, see Installing the NM-CIDS System Image, page 17-25.

Configuring IDS-Sensor Interfaces on the Router

NM-CIDS does not have an external console port. Console access to NM-CIDS is enabled when you issue the **service-module ids-module** *slot_number*/**0 session** command on the router, or when you initiate a Telnet connection into the router with the port number corresponding to the NM-CIDS slot. The lack of an external console port means that the initial bootup configuration is possible only through the router.

When you issue the **service-module ids-sensor** *slot_number*/**0 session** command, you create a console session with NM-CIDS, in which you can issue any IPS configuration commands. After completing work in the session and exiting the IPS CLI, you are returned to Cisco IOS CLI.

The **session** command starts a reverse Telnet connection using the IP address of the ids-sensor interface. The ids-sensor interface is an interface between NM-CIDS and the router. You must assign an IP address to the ids-sensor interface before invoking the **session** command. Assigning a routable IP address can make the ids-sensor interface itself vulnerable to attacks. To counter that vulnerability, a loopback IP address is assigned to the ids-sensor interface.

To configure the NM-CIDS interfaces, follow these steps:

Step 1 Confirm the NM-CIDS slot number in your router:

```
router # show interfaces ids-sensor slot_number/0
```

Note You can also use the **show run** command. Look for "IDS-Sensor" and the slot number.

Note Cisco IOS gives NM-CIDS the name "IDS-Sensor." In this example, 1 is the slot number and 0 is the port number, because there is only one port.

Step 2 Enable the CEF switching path:

```
router# configuration terminal
router(config)# ip cef
router(config)# exit
```

Step 3 Create a loopback interface:

```
router# configure terminal
router(config)# interface loopback 0
```

Step 4 Assign an IP address and netmask to the loopback interface:

```
router(config-if)# ip address 10.99.99.99 255.255.255.255
```

Note You must assign an IP address to the NM-CIDS's internal interface to session in to NM-CIDS. Choose a network that does not overlap with any networks assigned to the other interfaces in the router.

Step 5 Assign an unnumbered loopback interface to the ids-sensor interface. Use slot 1 for this example.

```
router(config)# interface ids-sensor 1/0
router(config-if)# ip unnumbered loopback 0
```

Step 6 Activate the port:

```
router(config-if)# no shutdown
```

Step 7 Exit configuration mode:

```
router(config-if)# end
```

Step 8 Write the configuration to NVRAM:

```
router# write memory
Building configuration
[OK]
```

Establishing NM-CIDS Sessions

This section describes how to establish sessions between the router and NM-CIDs. It contains the following topics:

- Sessioning to NM-CIDS, page 16-3
- Telneting to NM-CIDS, page 16-4

Sessioning to NM-CIDS

Use the **session** command to establish a session from the router to NM-CIDS. Press **Ctrl-Shift-6**, then **x**, to return a session prompt to a router prompt, that is, to go from the NM-CIDS prompt back to the router prompt. Press **Enter** on a blank line to go back to the session prompt, the NM-CIDS prompt. You should only suspend a session to NM-CIDS if you will be returning to the session after executing router commands. If you do not plan on returning to the NM-CIDS session, you should close the session rather than suspend it.

When you close a session, you are logged completely out of the NM-CIDS CLI and a new session connection requires a username and password to log in. A suspended session leaves you logged in to the CLI. When you connect with the **session** command, you can go back to the same CLI without having to provide your username and password.

Note Telnet clients vary. In some cases, you may have to press **Ctrl-6 + x**. The control character is specified as ^^, Ctrl-^, or ASCII value 30 (hex 1E).

Caution If you use the **disconnect** command to leave the session, the session remains running. The open session can be exploited by someone wanting to take advantage of a connection that is still in place.

To open and close sessions to NM-CIDS, follow these steps:

Step 1 Open a session from the router to NM-CIDS:

```
router# service-module ids-sensor 1/0 session
Trying 10.99.99.99, 2033 ... Open
```

Step 2 Exit the NM-CIDS session:

```
nm-cids# exit
```

Note If you are in submodes of the IPS CLI, you must exit all submodes. Type `exit` until the sensor login prompt appears.

Failing to close a session properly makes it possible for others to exploit a connection that is still in place. Remember to type **exit** at the `Router#` prompt to close the Cisco IOS session completely.

Step 3 Suspend and close the session to NM-CIDS by pressing **Ctrl-Shift** and pressing **6**. Release all keys, and then press **x**.

Note When you are finished with a session, you need to return to the router to establish the association between a session (the IPS application) and the router interfaces you want to monitor.

Step 4 Disconnect from the router:

```
router# disconnect
```

Step 5 Press **Enter** to confirm the disconnection:

```
router# Closing connection to 10.99.99.99 [confirm] <Enter>
```

Telneting to NM-CIDS

You can also Telnet directly to the router with the port number corresponding to the NM-CIDS slot. Use the address you established when configuring the loopback 0 interface in Configuring IDS-Sensor Interfaces on the Router, page 16-2. The port number is determined by the following formula: 2001 + 32 x slot number.

For example, for slot 1, the port number is 2033, for slot 2, it is 2065, and so forth.

To use Telnet to invoke a session to port 2033:

```
router# telnet 10.99.99.99 2033
```

Configuring Packet Capture

You must enable the desired interfaces (including subinterfaces) on the router for packet monitoring. You can select any number of interfaces or subinterfaces to be monitored. The packets sent and received on these interfaces are forwarded to NM-CIDS for inspection. You enable and disable the interfaces through the router CLI (Cisco IOS).

Note If the router is performing encryption, the NM-CIDS receives the packets after decryption coming into the router and before encryption leaving the router.

To configure packet capture on NM-CIDS, follow these steps:

Step 1 Log in to the router console.

Step 2 View your interface configuration:

```
router# show run
```

Step 3 Identify the interfaces or subinterfaces that you want to monitor, for example, FastEthernet0/0.

Note You can choose more than one interface or subinterface to monitor, but you can only edit one interface at a time.

Step 4 Enter global configuration mode:

```
router# configure terminal
```

Step 5 Specify the interface or subinterface:

```
router(config)# interface FastEthernet0/0
```

Note The traffic comes from one of the router interfaces.

Step 6 Configure the interface to copy network traffic to NM-CIDS:

```
router(config-if)# ids-service-module monitoring
```

Note Use the **no ids-service-module monitoring** command to turn off monitoring.

Step 7 Exit interface mode:

```
router(config-if)# exit
```

Step 8 Repeat Steps 3 though 6 for each interface or subinterface that you want to monitor.

Step 9 Exit global configuration mode:

```
router(config)# exit
```

Step 10 Verify that NM-CIDS is analyzing network traffic.

a. Open a Telnet or SSH session to the external interface on NM-CIDS, or establish a router console session.

Note SSH requires allowed hosts. For the procedure, see Adding Hosts to the Known Hosts List, page 4-32.

b. Log in to NM-CIDS.

c. View the interface statistics to make sure the monitoring interface is up:

```
nm-cids# show interface clear
nm-cids# show interface
MAC statistics from interface FastEthernet0/1
   Media Type = backplane
   Missed Packet Percentage = 0
   Inline Mode = Unpaired
   Pair Status = N/A
   Link Status = Up
   Link Speed = Auto_100
   Link Duplex = Auto_Full
   Total Packets Received = 23
   Total Bytes Received = 1721
   Total Multicast Packets Received = 0
   Total Broadcast Packets Received = 0
   Total Jumbo Packets Received = 0
   Total Undersize Packets Received = 0
   Total Receive Errors = 0
   Total Receive FIFO Overruns = 0
   Total Packets Transmitted = 2
   Total Bytes Transmitted = 120
   Total Multicast Packets Transmitted = 0
   Total Broadcast Packets Transmitted = 0
   Total Jumbo Packets Transmitted = 0
   Total Undersize Packets Transmitted = 0
   Total Transmit Errors = 0
   Total Transmit FIFO Overruns = 0
```

d. Repeat Step c to see the counters gradually increasing. This indicates that NM-CIDS is receiving network traffic.

If the counters are not increasing, make sure the you executed Steps 3 though 6 properly and that FastEthernet0/1 was added to the virtual sensor when you initialized the NM-CIDS with the **setup** command.

Administrative Tasks

The following section describes how to reboot NM-CIDS and how to check the status of the Cisco IPS software. It contains the following topics:

Shutting Down, Reloading, and Resetting NM-CIDS

The Cisco IOS provides the following commands to control NM-CIDS: **shutdown**, **reload**, and **reset**:

- **shutdown**—Brings the operating system down gracefully:

  ```
  router# service-module ids-sensor slot_number/0 shutdown
  ```

Caution Make sure you execute a **shutdown** command before you remove NM-CIDS from the router. Failing to do so can lead to the loss of data or the corruption of the hard-disk drive.

- **reload**—Performs a graceful halt and reboot of the operating system on NM-CIDS:

  ```
  router# service-module ids-sensor slot_number/0 reload
  ```

- **reset**—Resets the hardware on NM-CIDS. Typically this command is used to recover from a shutdown.

  ```
  router# service-module ids-sensor slot_number/0 reset
  ```

  The following warning appears:

  ```
  router# service-module ids-sensor 1/0 reset
  Use reset only to recover from shutdown or failed state
  Warning: May lose data on the hard disc!
  Do you want to reset?[confirm]
  ```

Caution Hard-disk drive data loss only occurs if you issue the **reset** command without first shutting down NM-CIDS. If NM-CIDS is still running correctly, use the **reload** command rather than the **reset** command. You can use the **reset** command safely in other situations.

Checking the Status of the Cisco IPS Software

Use the **status** command to check the status of the Cisco IPS software running on the router:

```
router# service-module ids-sensor slot_number/0 status
```

Something similar to the following output appears:

```
Router# service-module ids-sensor 1/0 status
Service Module is Cisco IDS-Sensor 1/0
Service Module supports session via TTY line 33
Service Module is in Steady state
Getting status from the Service Module, please wait..
Service Module Version information received,
Major ver = 1, Minor ver= 1
Cisco Systems Intrusion Detection System Network Module
Software version: 5.0(1)S42
Model: NM-CIDS
Memory: 254676 KB
Mgmt IP addr:     xx.xx.xx.xx
Mgmt web ports:    443
Mgmt TLS enabled:  true
```

Supported Cisco IOS Commands

The **service-module ids-sensor** *slot_number*/**0** Cisco IOS command is new to support NM-CIDS. The slot number can vary, but the port is always 0.

The following options apply:

- Privileged mode EXEC
  - **service-module ids-sensor** *slot_number*/**0 reload**

    Reloads the operating system on NM-CIDS.
  - **service-module ids-sensor** *slot_number*/**0 reset**

    Provides a hardware reset to NM-CIDS.
  - **service-module ids-sensor** *slot_number*/**0 session**

    The **session** command lets you access the IPS console.
  - **service-module ids-sensor** *slot_number*/**0 shutdown**

    Shuts down the IPS applications running on NM-CIDS.

Caution Removing the NM-CIDS without proper shutdown can result in the hard-disk drive being corrupted. After successful shutdown of the NM-CIDS applications, Cisco IOS prints a message indicating that you can now remove NM-CIDS.

  - **service-module ids-sensor** *slot_number*/**0 status**

    Provides information on the status of the Cisco IPS software.
- Configure interfaces mode `(config-if)`
  - **ids-service-module monitoring**

    You can enable IPS monitoring on a specified interface (or subinterface). Both inbound and outbound packets on the specified interface are forwarded for monitoring.

CHAPTER 17

Upgrading, Downgrading, and Installing System Images

Caution The BIOS on Cisco IDS/IPS sensors is specific to Cisco IDS/IPS sensors and must only be upgraded under instructions from Cisco with BIOS files obtained from the Cisco website. Installing a non-Cisco or third-party BIOS on Cisco IDS/IPS sensors voids the warranty. For more information on how to obtain instructions and BIOS files from the Cisco website, see Obtaining Cisco IPS Software, page 18-1.

This chapter describes how to upgrade, downgrade, and install system images. It contains the following sections:

Overview

Note After you upgrade any IPS software on your sensor, you must restart the IDM to see the latest software features.

You can upgrade and downgrade the software on the sensor. Upgrading applies a service pack, signature update, minor version, major version, or recovery partition file. Downgrading removes the last applied upgrade from the sensor.

Caution You cannot use the **downgrade** command to go from 5.*x* to 4.*x*. To revert to 4.*x*, you must reimage the sensor.

You can recover the application partition image on your sensor if it becomes unusable. Using the **recover** command lets you retain your host settings while other settings revert to the factory defaults.

To install a new system image on the sensor, use the recovery/upgrade CD, ROMMON, the bootloader/helper file, or the maintenance partition depending on which platform you have.

When you install a new system image on your sensor, all accounts are removed and the default cisco account is reset to use the default password **cisco**. After installing the system image, you must initialize the sensor again. For the procedure, see Initializing the Sensor, page 3-2.

After you reimage and initialize your sensor, upgrade your sensor with the most recent service pack, signature update, minor version, major version, and recovery partition file. For the procedure for locating software on Cisco.com, see Obtaining Cisco IPS Software, page 18-1.

Upgrading the Sensor

This section explains how to use the **upgrade** command to upgrade the software on the sensor. It contains the following topics:

Overview

You can upgrade the sensor with the following files, all of which have the extension .pkg:

Note Upgrading the sensor changes the software version of the sensor.

Cisco IPS 5.1(1) through 5.1(4):

- Signature updates, for example, IPS-sig-S150-minreq-5.1-1.pkg
- Major updates, for example, IPS-K9-maj-6.0-1-pkg
- Minor updates, for example, IPS-K9-min-5.1-1.pkg
- Service packs, for example, IPS-K9-sp-5.1-2.pkg
- Recovery packages, for example, IPS-K9-r-1.1-a-5.1-1.pkg

Cisco IPS 5.1(5)E1 and later:

- Signature updates, for example, IPS-sig-S700-req-E1.pkg
- Signature engine updates, for example, IPS-engine-E1-req-5.1-3.pkg
- Major updates, for example, IPS-K9-5.1-1-E1.pkg
- Minor updates, for example, IPS-K9-5.1-1-E1.pkg
- Service packs, for example, IPS-K9-5.1-3-E1.pkg
- Patch releases, for example, IPS-K9-patch-5.1-1pl-E1.pkg
- Recovery packages, for example, IPS-K9-r-1.1-a-5.1-1-E1.pkg

Upgrade Command and Options

Use the **auto-upgrade-option enabled** command in the service host submode to configure automatic upgrades.

The following options apply:

- **default**— Sets the value back to the system default setting.
- **directory**— Directory where upgrade files are located on the file server.

  A leading '/' indicates an absolute path.
- **file-copy-protocol**— File copy protocol used to download files from the file server. The valid values are **ftp** or **scp**.

  Note If you use SCP, you must use the **ssh host-key** command to add the server to the SSH known hosts list so the sensor can communicate with it through SSH. For the procedure, see Adding Hosts to the Known Hosts List, page 4-32.
- **ip-address**— IP address of the file server.
- **password**— User password for authentication on the file server.
- **schedule-option**—Schedules when automatic upgrades occur. Calendar scheduling starts upgrades at specific times on specific days. Periodic scheduling starts upgrades at specific periodic intervals.
  - **calendar-schedule**—Configure the days of the week and times of day that automatic upgrades will be performed.

    days-of-week—Days of the week on which auto-upgrades will be performed. You can select multiple days: *sunday* through *saturday* are the valid values.

    no—Removes an entry or selection setting.

    times-of-day—Times of day at which auto-upgrades will begin. You can select multiple times. The valid value is *hh:mm[:ss]*.
  - **periodic-schedule**—Configure the time that the first automatic upgrade should occur, and how long to wait between automatic upgrades.

    interval—The number of hours to wait between automatic upgrades. Valid values are 0 to 8760.

    start-time—The time of day to start the first automatic upgrade. The valid value is *hh:mm[:ss]*.
- **user-name**—Username for authentication on the file server.

Using the Upgrade Command

To upgrade the sensor, follow these steps:

Step 1 Download the major update file (for example, IPS-K9-6.0-2-E1.pkg) to an FTP, SCP, HTTP, or HTTPS server that is accessible from your sensor.

Note You must log in to Cisco.com using an account with cryptographic privileges to download the file. Do not change the file name. You must preserve the original file name for the sensor to accept the update. For the procedure for obtaining software on Cisco.com and an account with cryptographic privileges, see Obtaining Cisco IPS Software, page 18-1.

Step 2 Log in to the CLI using an account with administrator privileges.

Step 3 Enter configuration mode:

```
sensor# configure terminal
```

Step 4 Upgrade the sensor:

```
sensor# configure terminal
sensor(config)# upgrade scp://tester@10.1.1.1//upgrade/IPS-K9-6.0-2-E1.pkg
```

Step 5 Enter the password when prompted:

```
Enter password: ********
Re-enter password: ********
```

Step 6 Type **yes** to complete the upgrade.

Note Major and minor updates and service packs may force a restart of the IPS processes or even force a reboot of the sensor to complete installation.

Step 7 Verify your new sensor version:

```
sensor# show version
Application Partition:

Cisco Intrusion Prevention System, Version 6.0(2)E1

Host:
    Realm Keys          key1.0
Signature Definition:
    Signature Update    S280.0                  2007-04-11
    Virus Update        V1.2                    2005-11-24
OS Version:             2.4.30-IDS-smp-bigphys
Platform:               IPS-4260
Serial Number:          AZBW5470042
No license present
Sensor up-time is 2 days.
Using 1897000960 out of 3569864704 bytes of available memory (53% usage)
system is using 17.8M out of 29.0M bytes of available disk space (61% usage)
application-data is using 43.8M out of 166.8M bytes of available disk space (28%
 usage)
boot is using 37.9M out of 69.5M bytes of available disk space (57% usage)


MainApp          2007_MAR_29_14_06   (Release)   2007-03-29T14:44:36-0600   Running
```

```
AnalysisEngine   2007_MAR_29_14_06   (Release)   2007-03-29T14:44:36-0600   Running
CLI              2007_MAR_29_14_06   (Release)   2007-03-29T14:44:36-0600


Upgrade History:

  IPS-K9-6.0-2-E1   14:06:00 UTC Thu Mar 29 2007

Recovery Partition Version 1.1 6.0(2)E1

sensor#
```

Upgrading the Recovery Partition

Use the **upgrade** command to upgrade the recovery partition with the most recent version so that it is ready if you need to recover the application partition on your sensor.

Note Recovery partition images are generated for major and minor software releases and only in rare situations for service packs or signature updates.

Note To upgrade the recovery partition the sensor must already be running version 5.0(1) or later.

To upgrade the recovery partition on your sensor, follow these steps:

Step 1 Download the recovery partition image file (IPS-K9-r-1.1-a-5.0-1-E1.pkg) to an FTP, SCP, HTTP, or HTTPS server that is accessible from your sensor.

For the procedure for locating software on Cisco.com, see Obtaining Cisco IPS Software, page 18-1.

Caution Some browsers add an extension to the filename. The filename of the saved file must match what is displayed on the download page or you cannot use it to upgrade the recovery partition.

Step 2 Log in to the CLI using an account with administrator privileges.

Step 3 Enter configuration mode:

```
sensor# configure terminal
```

Step 4 Upgrade the recovery partition:

```
sensor(config)#
upgrade scp://user@server_ipaddress//upgrade_path/IPS-K9-r-1.1-a-5.0-1-E1.pkg

sensor(config)#
upgrade ftp://user@server_ipaddress//upgrade_path/IPS-K9-r-1.1-a-5.0-1-E1.pkg
```

Step 5 Type the server password.

The upgrade process begins.

Note This procedure only reimages the recovery partition. The application partition is not modified by this upgrade. To reimage the application partition after the recovery partition, use the **recover application-partition** command. For the procedure, see Using the Recover Command, page 17-11.

Configuring Automatic Upgrades

This section describes how to configure the sensor to automatically look for upgrades in the upgrade directory. It contains the following topics:

Overview

You can configure the sensor to look for new upgrade files in your upgrade directory automatically.

You must download the software upgrade from Cisco.com and copy it to the upgrade directory before the sensor can poll for automatic upgrades. For the procedure for locating software on Cisco.com see Obtaining Cisco IPS Software, page 18-1.

Auto-upgrade Command and Options

Use the **auto-upgrade-option enabled** command in the service host submode to configure automatic upgrades.

The following options apply:

- **default**— Sets the value back to the system default setting.
- **directory**— Directory where upgrade files are located on the file server.

  A leading '/' indicates an absolute path.
- **file-copy-protocol**— File copy protocol used to download files from the file server. The valid values are **ftp** or **scp**.

  Note If you use SCP, you must use the **ssh host-key** command to add the server to the SSH known hosts list so the sensor can communicate with it through SSH. For the procedure, see Adding Hosts to the Known Hosts List, page 4-32.
- **ip-address**— IP address of the file server.
- **password**— User password for authentication on the file server.

- **schedule-option**—Schedules when automatic upgrades occur. Calendar scheduling starts upgrades at specific times on specific days. Periodic scheduling starts upgrades at specific periodic intervals.
  - **calendar-schedule**—Configure the days of the week and times of day that automatic upgrades will be performed.

    days-of-week—Days of the week on which auto-upgrades will be performed. You can select multiple days: *sunday* through *saturday* are the valid values.

    no—Removes an entry or selection setting.

    times-of-day—Times of day at which auto-upgrades will begin. You can select multiple times. The valid value is *hh:mm[:ss]*.
  - **periodic-schedule**—Configure the time that the first automatic upgrade should occur, and how long to wait between automatic upgrades.

    interval—The number of hours to wait between automatic upgrades. Valid values are 0 to 8760.

    start-time—The time of day to start the first automatic upgrade. The valid value is *hh:mm[:ss]*.
- **user-name**—Username for authentication on the file server.

Using the auto-upgrade Command

To schedule automatic upgrades, follow these steps:

Step 1 Log in to the CLI using an account with administrator privileges.

Step 2 Configure the sensor to automatically look for new upgrades in your upgrade directory.

```
sensor# configure terminal
sensor(config)# service host
sensor(config-hos)# auto-upgrade-option enabled
```

Step 3 Specify the scheduling:

a. For calendar scheduling, which starts upgrades at specific times on specific day:

```
sensor(config-hos-ena)# schedule-option calendar-schedule
sensor(config-hos-ena-cal# days-of-week sunday
sensor(config-hos-ena-cal# times-of-day 12:00:00
```

b. For periodic scheduling, which starts upgrades at specific periodic intervals:

```
sensor(config-hos-ena)# schedule-option periodic-schedule
sensor(config-hos-ena-per)# interval 24
sensor(config-hos-ena-per)# start-time 13:00:00
```

Step 4 Specify the IP address of the file server:

```
sensor(config-hos-ena-per)# exit
sensor(config-hos-ena)# ip-address 10.1.1.1
```

Step 5 Specify the directory where the upgrade files are located on the file server:

```
sensor(config-hos-ena)# directory /tftpboot/update/5.1_dummy_updates
```

Step 6 Specify the username for authentication on the file server:

```
sensor(config-hos-ena)# user-name tester
```

Step 7 Specify the password of the user:

```
sensor(config-hos-ena)# password
Enter password[]: ******
Re-enter password: ******
```

Step 8 Specify the file server protocol:

```
sensor(config-hos-ena)# file-copy-protocol ftp
```

Note If you use SCP, you must use the **ssh host-key** command to add the server to the SSH known hosts list so the sensor can communicate with it through SSH. For the procedure, see Adding Hosts to the Known Hosts List, page 4-32.

Step 9 Verify the settings:

```
sensor(config-hos-ena)# show settings
   enabled
   ----------------------------------------------
      schedule-option
      ---------------------------------------------
         periodic-schedule
         ---------------------------------------------
            start-time: 13:00:00
            interval: 24 hours
         ---------------------------------------------
      ---------------------------------------------
      ip-address: 10.1.1.1
      directory: /tftpboot/update/5.0_dummy_updates
      user-name: tester
      password: <hidden>
      file-copy-protocol: ftp default: scp
   ----------------------------------------------
sensor(config-hos-ena)#
```

Step 10 Exit auto upgrade submode:

```
sensor(config-hos-ena)# exit
sensor(config-hos)# exit
Apply Changes:?[yes]:
```

Step 11 Press **Enter** to apply the changes or type **no** to discard them.

UNIX-Style Directory Listings

To configure Auto Update using an FTP server, the FTP server must provide directory listing responses in UNIX style. MS-DOS style directory listing is not supported by the sensor Auto Update feature.

Note If the server supplies MS-DOS style directory listings, the sensor cannot parse the directory listing and does not know that there is a new update available.

To change Microsoft IIS to use UNIX-style directory listings, follow these steps:

Step 1 **Choose Start > Program Files > Administrative Tools**.

Step 2 Click the **Home Directory** tab.

Step 3 Click the **UNIX directory listings style** radio button.

Automatic Upgrade Examples

Table 17-1 shows automatic upgrade examples. In these examples, the upgrades are configured hourly starting at 1:00. For example, Cycle 1 begins at 1:00, Cycle 2 begins at 2:00, and Cycle 3 begins at 3:00.

Table 17-1 Automatic Upgrade Example Cases

Case/Current Version	Files in Remote Directory	Automatic Update Cycle/New Version
Case 0 5.1(4) E0 S250	• IPS-sig-S260-minreq-5.0-6.pkg • IPS-engine-E2-req-5.1-4.pkg • IPS-sig-S262-req-E2.pkg • IPS-sig-S263-req-E2.pkg • IPS-engine-E3-req-5.1-4.pkg • IPS-sig-S264-req-E3.pkg	• Cycle 1 installs IPS-engine-E3-req-5.1-4.pkg. New version is 5.1(4) E2 S250. • Cycle 2 installs IPS-sig-S264-req-E3.pkg. New version is 5.1(4) E2 S264.
Case 1 5.1(4) E0 S250	• IPS-K9-sp-5.1-5.pkg • IPS-sig-S260-minreq-5.0-6.pkg • IPS-K9-5.1-6-E1.pkg • IPS-engine-E2-req-5.1-6.pkg • IPS-sig-S262-req-E2.pkg • IPS-sig-S263-req-E2.pkg	• Cycle 1 installs IPS-K9-5.1-6-E1.pkg. New version is 5.1(6) E1 S260. • Cycle 2 installs IPS-engine-E2-req-5.1-6.pkg. New version is 5.1(6) E2 S260. • Cycle 3 installs IPS-sig-S263-req-E2.pkg. New version is 5.1(6) E2 S263.
Case 2 5.1(6) E5 S300	• IPS-K9-6.0-1-E7.pkg • IPS-K9-6.0-2-E9.pkg • IPS-K9-6.0-3-E11.pkg • IPS-engine-E10-req-6.0-2.pkg • IPS-engine-E12-req-6.0-3.pkg • IPS-sig-S305-req-E12.pkg • IPS-sig-S307-req-E12.pkg	• Cycle 1 installs IPS-K9-6.0-3-E11.pkg. New version is 6.0(3) E11 S300. • Cycle 2 installs IPS-engine-E12-req-6.0-3.pkg. New version is 6.0(3) E12 S300. • Cycle 3 installs IPS-sig-S307-req-E12.pkg. New version is 6.0(3) E12 S307.
Case 3 5.1(6) E10 S300	• IPS-K9-6.0-1-E9.pkg • IPS-engine-E11-req-6.0-1.pkg • IPS-sig-S305-req-E11.pkg • IPS-sig-S307-req-E11.pkg	• Cycle 1 installs nothing because E9 is less than E10.

Table 17-1 Automatic Upgrade Example Cases (continued)

Case/Current Version	Files in Remote Directory	Automatic Update Cycle/New Version
Case 4 5.1(6) E10 S300	• IPS-engine-E11-req-5.1-6.pkg • IPS-sig-S301-req-E10.pkg	• Cycle 1 installs IPS-engine-E11-req-5.1-6.pkg. New version is 5.1(6) E11 S300.
Case 5 5.1(6) E10 S300	• IPS-sig-S301-req-E10.pkg • IPS-sig-S302-req-E11.pkg • IPS-sig-S303-req-E12.pkg • IPS-engine-E11-req-5.1-6.pkg	• Cycle 1 installs IPS-engine-E11-req-5.1-6.pkg. New version is 5.1(6) E11 S300. • Cycle 2 installs IPS-sig-S302-req-E11.pkg. New version is 5.1(6) E11 S302.
Case 6 6.0(3)E1 S300 (IPS 4270-20)	• IPS-K9-6.0-4-E1.pkg • IPS-4270_20-K9-6.0-4-E1.pkg	• Cycle 1 installs IPS-4270_20-K9-6.0-4-E1.pkg New version is 6.0(4)E1 S310
Case 7 6.0(4)E3 S330 (AIM-IPS)	• IPS-K9-6.0-5-E3.pkg • IPS-AIM-K9-6.0-5-E3.pkg	• Cycle 1 installs IPS-AIM-K9-6.0-5-E3.pkg. New version is 6.0(5)E3 S335.
Case 8 6.0(5)E5 S330 (AIM-IPS)	• IPS-K9-7.0-1-E5.pkg • IPS-AIM-K9-7.0-1-E5.pkg	• Cycle 1 installs IPS-K9-7.0-1-E5.pkg. New version is 7.0(1)E5 S377

Downgrading the Sensor

Use the **downgrade** command to remove the last applied upgrade from the sensor.

Caution You cannot use the **downgrade** command to go from 5.*x* to 4.*x*. To revert to 4.*x*, you must reimage the sensor.

To remove the last applied upgrade from the sensor, follow these steps:

Step 1 Log in to the sensor using an account with administrator privileges.

Step 2 Enter global configuration mode:

```
sensor# configure terminal
```

Step 3 Downgrade the sensor:

```
sensor(config)# downgrade
Warning: Executing this command will reboot the system and downgrade to
IPS-K9-sp.5.0-2.pkg. Configuration changes made since the last upgrade will be lost and
the system may be rebooted.
Continue with downgrade?:
```

Step 4 Type **yes** to continue with the downgrade.

Step 5 If there is no recently applied service pack or signature update, the **downgrade** command is not available:

```
sensor(config)# downgrade
No downgrade available.
sensor(config)#
```

Recovering the Application Partition

This section explains how to recover the application partition, and contains the following topics:

- Overview, page 17-11
- Using the Recover Command, page 17-11

Overview

You can recover the application partition image for the appliance if it becomes unusable. Some network configuration information is retained when you use this method, which lets you have network access after the recovery is performed.

Use the **recover application-partition** command to boot to the recovery partition, which automatically recovers the application partition on your appliance.

Note If you have upgraded your recovery partition to the most recent version before you recover the application partition image, you can install the most up-to-date software image. For the procedure for upgrading the recovery partition to the most recent version, see Using the Recover Command, page 17-11.

Because you can execute the **recover application-partition** command through a Telnet or SSH connection, we recommend using this command to recover sensors that are installed at remote locations.

Note If the appliance supports it, you can also use the recovery/upgrade CD to reinstall both the recovery and application partitions. For the procedure, see Using the Recovery/Upgrade CD, page 17-23.

Note When you reconnect to the sensor after recovery, you must log in with the default username and password `cisco`.

Using the Recover Command

To recover the application partition image, follow these steps:

Step 1 Download the recovery partition image file (IPS-K9-r-1.1-a-5.0-1-E1.pkg) to the tftp root directory of a TFTP server that is accessible from your sensor. For the procedure for locating software on Cisco.com, see Obtaining Cisco IPS Software, page 18-1.

Note Make sure you can access the TFTP server location from the network connected to the Ethernet port of your sensor.

Step 2 Log in to the CLI using an account with administrator privileges.

Step 3 Enter configuration mode:

```
sensor# configure terminal
```

Step 4 Recover the application partition image:

```
sensor(config)# recover application-partition
Warning: Executing this command will stop all applications and re-image the node to
version 5.0(0.27)S91(0.27). All configuration changes except for network settings will be
reset to default.
Continue with recovery? []:
```

Step 5 Type **yes** to continue.

Shutdown begins immediately after you execute the **recover** command. Shutdown can take a while, and you will still have access to the CLI, but access will be terminated without warning.

The application partition is reimaged using the image stored on the recovery partition. You must now initialize the appliance with the **setup** command. For the procedure, see Initializing the Sensor, page 3-2.

Note The IP address, netmask, access lists, time zone, and offset are saved and applied to the reimaged application partition. If you executed the **recover application-partition** command remotely, you can SSH to the sensor with the default username and password (cisco/cisco) and then initialize the sensor again with the **setup** command. You cannot use Telnet until you initialize the sensor because Telnet is disabled by default.

If you cannot access the CLI to execute the **recover application-partition** command, you can reboot the sensor and select the option from the boot menu during the bootup process. This lets you boot to the recovery partition and reimage the application partition. Executing the **recovery** command in this way requires console or keyboard and monitor access to the sensor, which is possible on the appliances and NM-CIDS, but not on IDSM-2 or AIP-SSM.

Installing System Images

This section contains the procedures for installing system images on the appliances and modules. It contains the following topics:

- Using the Recovery/Upgrade CD, page 17-23
- Installing the NM-CIDS System Image, page 17-25
- Installing the IDSM-2 System Image, page 17-27
- Installing the AIP-SSM System Image, page 17-38

Caution All user configuration settings are lost when you install the system image. Before trying to recover the sensor by installing the system image, try to recover by using the **recover application-partition** command or by selecting the recovery partition during sensor bootup. For the procedure see Recovering the Application Partition, page 17-11.

Understanding ROMMON

Some Cisco sensors include a preboot CLI called ROMMON, which lets you boot images on sensors where the image on the primary device is missing, corrupt, or otherwise unable to boot the normal application. ROMMON is particularly useful for recovering remote sensors as long as the serial console port is available.

Access to ROMMON is available only through the serial console port, a Cisco-standard asynchronous RS-232C DTE available in an RJ-45F connector on the sensor chassis. The serial port is configured for 9600 baud, 8 data bits, 1 stop bit, no parity, and no flow control. For the procedure for using a terminal server, see Connecting an Appliance to a Terminal Server, page 17-13.

TFTP Servers

ROMMON uses TFTP to download an image and launch it. TFTP does not address network issues such as latency or error recovery. It does implement a limited packet integrity check so that packets arriving in sequence with the correct integrity value have an extremely low probability of error. But TFTP does not offer pipelining so the total transfer time is equal to the number of packets to be transferred times the network average RTT. Because of this limitation, we recommend that the TFTP server be located on the same LAN segment as the sensor. Any network with an RTT less than a 100 milliseconds should provide reliable delivery of the image. Be aware that some TFTP servers limit the maximum file size that can be transferred to ~32 MB.

Connecting an Appliance to a Terminal Server

A terminal server is a router with multiple, low speed, asynchronous ports that are connected to other serial devices. You can use terminal servers to remotely manage network equipment, including appliances.

To set up a Cisco terminal server with RJ-45 or hydra cable assembly connections, follow these steps:

Step 1 Connect to a terminal server using one of the following methods:

- For IDS-4215, IPS-4240, and IPS-4255:
  - For RJ-45 connections, connect a 180 rollover cable from the console port on the appliance to a port on the terminal server.

  - For hydra cable assemblies, connect a straight-through patch cable from the console port on the appliance to a port on the terminal server.
- For all other appliances, connect the M.A.S.H. adapter (part number 29-4077-01) to COM1 on the appliance and:
  - For RJ-45 connections, connect a 180 rollover cable from the M.A.S.H. adapter to a port on the terminal server.
  - For hydra cable assemblies, connect a straight-through patch cable from the M.A.S.H. adapter to a port on the terminal server.

Step 2 Configure the line and port on the terminal server as follows:

a. In enable mode, enter the following configuration, where # is the line number of the port to be configured:

```
config t
line #
login
transport input all
stopbits 1
flowcontrol hardware
speed 9600
exit
exit
wr mem
```

b. If you are configuring a terminal server for an IDS-4215, IPS-4240, or IPS-4255, go to Step 3.

Otherwise, for all other supported appliances, to direct all output to the terminal server, log in to the CLI and enter the following commands:

```
sensor# configure terminal
sensor(config)# display-serial
```

Output is directed to the serial port. Use the **no display-serial** command to redirect output to the keyboard and monitor.

Note You can set up a terminal server and use the **display-serial** command to direct all output from the appliance to the serial port. This option lets you view system messages on a console connected to the serial port, even during the boot process. When you use this option, all output is directed to the serial port and any local keyboard and monitor connection is disabled. However, BIOS and POST messages are still displayed on the local keyboard and monitor. For the procedure, see Directing Output to a Serial Connection, page 13-21.

Note There are no keyboard or monitor ports on an IDS-4215, IPS-4240, or IPS-4255; therefore, the **display-serial** and **no display-serial** commands do not apply to those platforms.

Step 3 Be sure to properly close a terminal session to avoid unauthorized access to the appliance.

If a terminal session is not stopped properly, that is, if it does not receive an exit(0) signal from the application that initiated the session, the terminal session can remain open. When terminal sessions are not stopped properly, authentication is not performed on the next session that is opened on the serial port.

Caution Always exit your session and return to a login prompt before terminating the application used to establish the connection.

Caution If a connection is dropped or terminated by accident, you should reestablish the connection and exit normally to prevent unauthorized access to the appliance.

Installing the IDS-4215 System Image

You can install the IDS-4215 system image by using the ROMMON on the appliance to TFTP the system image onto the compact flash device.

Caution Before installing the system image, you must first upgrade the IDS-4215 BIOS to version 5.1.7 and the ROMMON to version 1.4 using the upgrade utility file IDS-4215-bios-5.1.7-rom-1.4.bin. For the procedure, see Upgrading the IDS-4215 BIOS and ROMMON, page 17-17.

To install the IDS-4215 system image, follow these steps:

Step 1 Download the IDS-4215 system image file (IPS-4215-K9-sys-1.1-a-5.1-5-E1.img) to the tftp root directory of a TFTP server that is accessible from your IDS-4215.

For the procedure for locating software on Cisco.com, see Obtaining Cisco IPS Software, page 18-1.

Make sure you can access the TFTP server location from the network connected to your IDS-4215 Ethernet port.

Step 2 Boot IDS-4215.

Step 3 Press **Ctrl-R** at the following prompt while the system is booting:

```
Evaluating Run Options...
```

Note You have five seconds to press **Ctrl-R**.

The console display resembles the following:

```
CISCO SYSTEMS IDS-4215
Embedded BIOS Version 5.1.7 02/23/04 15:50:39.31
Compiled by dnshep
Evaluating Run Options ...
Cisco ROMMON (1.4) #3: Mon Feb 23 15:52:45 MST 2004
Platform IDS-4215

Image Download Memory Sizing
Available Image Download Space: 510MB

0: i8255X @ PCI(bus:0 dev:13 irq:11)
1: i8255X @ PCI(bus:0 dev:14 irq:11)

Using 1: i82557 @ PCI(bus:0 dev:14 irq:11), MAC: 0000.0001.0001
Use ? for help.
rommon>
```

Step 4 Verify that IDS-4215 is running BIOS version 5.1.7 or later and ROMMON version 1.4 or later.

Note If IDS-4215 does not have the correct BIOS and ROMMON versions, you must upgrade the BIOS and ROMMON before reimaging. For the procedure, see Upgrading the IDS-4215 BIOS and ROMMON, page 17-17.

The current versions are shown in the console display information identified in Step 3.

Step 5 If necessary, change the port used for the TFTP download:

```
rommon> interface port_number
```

The port in use is listed just before the rommon prompt. In the example, port 1 is being used as noted by the text, `Using 1: i82557 @ PCI(bus:0 dev:14 irq:11), MAC: 0000.0001.0001`.

Note The default port used for TFTP downloads is port 1, which corresponds with the command and control interface of IDS-4215.

Note Ports 0 (monitoring interface) and 1 (command and control interface) are labeled on the back of the chassis.

Step 6 Specify an IP address for the local port on IDS-4215:

```
rommon> address ip_address
```

Note Use the same IP address that is assigned to IDS-4215.

Step 7 Specify the TFTP server IP address:

```
rommon> server ip_address
```

Step 8 Specify the gateway IP address:

```
rommon> gateway ip_address
```

Step 9 Verify that you have access to the TFTP server by pinging it from the local Ethernet port:

```
rommon> ping server_ip_address
rommon> ping server
```

Step 10 Specify the path and filename on the TFTP file server from which you are downloading the image:

```
rommon> file path/filename
```

UNIX example:

```
rommon> file /system_images/IPS-4215-K9-sys-1.1-a-5.1-5-E1.img
```

Note The path is relative to the UNIX TFTP server's default tftpboot directory. Images located in the default tftpboot directory do not have any directory names or slashes in the file location.

Windows example:

```
rommon> file C:\tftp_directory\IPS-4215-K9-sys-1.1-a-5.1-5-E1.img
```

Step 11 Download and install the system image:

```
rommon> tftp
```

Note IDS-4215 reboots several times during the reimaging process. Do not remove power from IDS-4215 during the update process or the upgrade can become corrupted.

Upgrading the IDS-4215 BIOS and ROMMON

The BIOS/ROMMON upgrade utility (IDS-4215-bios-5.1.7-rom-1.4.bin) upgrades the BIOS of IDS-4215 to version 5.1.7 and the ROMMON to version 1.4.

To upgrade the BIOS and ROMMON on IDS-4215, follow these steps:

Step 1 Download the BIOS ROMMON upgrade utility (IDS-4215-bios-5.1.7-rom-1.4.bin) to the TFTP root directory of a TFTP server that is accessible from IDS-4215.

For the procedure for locating software on Cisco.com, see Obtaining Cisco IPS Software, page 18-1.

Note Make sure you can access the TFTP server location from the network connected to the Ethernet port of IDS-4215.

Step 2 Boot IDS-4215. While rebooting, IDS-4215 runs the BIOS POST. After the completion of POST, the console displays the message: `Evaluating Run Options ...` for about 5 seconds.

Step 3 Press **Ctrl-R** while this message is displayed to display the ROMMON menu.

The console display resembles the following:

```
CISCO SYSTEMS IDS-4215
Embedded BIOS Version 5.1.3 05/12/03 10:18:14.84
Compiled by ciscouser
Evaluating Run Options ...
Cisco ROMMON (1.2) #0: Mon May 12 10:21:46 MDT 2003
Platform IDS-4215
0: i8255X @ PCI(bus:0 dev:13 irq:11)
1: i8255X @ PCI(bus:0 dev:14 irq:11)
Using 1: i82557 @ PCI(bus:0 dev:14 irq:11), MAC: 0000.c0ff.ee01
Use ? for help.
rommon>
```

Step 4 If necessary, change the port number used for the TFTP download:

```
rommon> interface port_number
```

The port in use is listed just before the rommon prompt. Port 1 (default port) is being used as indicated by the text, `Using 1: i82557 @ PCI(bus:0 dev:14 irq:11), MAC: 0000.c0ff.ee01`.

Note Ports 0 (monitoring port) and 1 (command and control port) are labeled on the back of the chassis.

Step 5 Specify an IP address for the local port on IDS-4215:

```
rommon> address ip_address
```

Note Use the same IP address that is assigned to IDS-4215.

Step 6 Specify the TFTP server IP address:

```
rommon> server ip_address
```

Step 7 Specify the gateway IP address:

```
rommon> gateway ip_address
```

Step 8 Verify that you have access to the TFTP server by pinging it from the local Ethernet port:

```
rommon> ping server_ip_address
rommon> ping server
```

Step 9 Specify the filename on the TFTP file server from which you are downloading the image:

```
rommon> file filename
```

Example:

```
rommon> file IDS-4215-bios-5.1.7-rom-1.4.bin
```

Note The syntax of the file location depends on the type of TFTP server used. Contact your system or network administrator for the appropriate syntax if the above format does not work.

Step 10 Download and run the update utility:

```
rommon> tftp
```

Step 11 Type **y** at the upgrade prompt and the update is executed.

IDS-4215 reboots when the update is complete.

Caution Do not remove power to IDS-4215 during the update process, otherwise the upgrade can get corrupted. If this occurs, IDS-4215 will be unusable and require an RMA.

Installing the IPS-4240 and IPS-4255 System Image

You can install the IPS-4240 and IPS-4255 system image by using the ROMMON on the appliance to TFTP the system image onto the compact flash device.

Note This procedure is for IPS-4240, but is also applicable to IPS-4255. The system image for IPS-4255 has “4255” in the filename.

To install the IPS-4240 and IPS-4255 system image, follow these steps:

Step 1 Download the IPS-4240 system image file (IPS-4240-K9-sys-1.1-a-5.1-5-E1.img) to the tftp root directory of a TFTP server that is accessible from your IPS-4240. For the procedure for locating software on Cisco.com, see Obtaining Cisco IPS Software, page 18-1.

Note Make sure you can access the TFTP server location from the network connected to the Ethernet port of your IPS-4240.

Step 2 Boot IPS-4240. The console display resembles the following:

```
Booting system, please wait...

CISCO SYSTEMS
Embedded BIOS Version 1.0(5)0 09/14/04 12:23:35.90

Low Memory: 631 KB
High Memory: 2048 MB
PCI Device Table.
Bus Dev Func VendID DevID Class              Irq
00  00  00   8086   2578  Host Bridge
00  01  00   8086   2579  PCI-to-PCI Bridge
00  03  00   8086   257B  PCI-to-PCI Bridge
00  1C  00   8086   25AE  PCI-to-PCI Bridge
00  1D  00   8086   25A9  Serial Bus         11
00  1D  01   8086   25AA  Serial Bus         10
00  1D  04   8086   25AB  System
00  1D  05   8086   25AC  IRQ Controller
00  1D  07   8086   25AD  Serial Bus         9
00  1E  00   8086   244E  PCI-to-PCI Bridge
00  1F  00   8086   25A1  ISA Bridge
00  1F  02   8086   25A3  IDE Controller     11
00  1F  03   8086   25A4  Serial Bus         5
00  1F  05   8086   25A6  Audio              5
02  01  00   8086   1075  Ethernet           11
03  01  00   177D   0003  Encrypt/Decrypt    9
03  02  00   8086   1079  Ethernet           9
03  02  01   8086   1079  Ethernet           9
03  03  00   8086   1079  Ethernet           9
03  03  01   8086   1079  Ethernet           9
04  02  00   8086   1209  Ethernet           11
04  03  00   8086   1209  Ethernet           5

Evaluating BIOS Options ...
Launch BIOS Extension to setup ROMMON

Cisco Systems ROMMON Version (1.0(5)0) #1: Tue Sep 14 12:20:30 PDT 2004

Platform IPS-4240-K9
Management0/0

MAC Address: 0000.c0ff.ee01
```

Step 3 Press **Break** or **Esc** at the following prompt while the system is booting to interrupt boot. Press the spacebar to begin boot immediately.

Note You have ten seconds to press **Break** or **Esc**.

```
Use BREAK or ESC to interrupt boot.
Use SPACE to begin boot immediately.
```

The system enters ROMMON mode. The `rommon>` prompt appears.

Step 4 Check the current network settings:

```
rommon> set
```

The output on the configured system resembles the following:

```
ROMMON Variable Settings:
      ADDRESS=0.0.0.0
      SERVER=0.0.0.0
      GATEWAY=0.0.0.0
      PORT=Management0/0
      VLAN=untagged
      IMAGE=
      CONFIG=
```

The variables have the following definitions:

- Address—Local IP address of IPS-4240
- Server—TFTP server IP address where the application image is stored
- Gateway—Gateway IP address used by IPS-4240
- Port—Ethernet interface used for IPS-4240 management
- VLAN—VLAN ID number (leave as untagged)
- Image—System image file/path name
- Config—Unused by these platforms

Note Not all values are required to establish network connectivity. The address, server, gateway, and image values are required. If you are not sure of the settings needed for your local environment, contact your system administrator.

Step 5 If necessary, change the interface used for the TFTP download:

Note The default interface used for TFTP downloads is Management0/0, which corresponds to the MGMT interface of IPS-4240.

```
rommon> PORT=interface_name
```

Step 6 If necessary, assign an IP address for the local port on IPS-4240:

```
rommon> ADDRESS=ip_address
```

Note Use the same IP address that is assigned to IPS-4240.

Step 7 If necessary, assign the TFTP server IP address:

```
rommon> SERVER=ip_address
```

Step 8 If necessary, assign the gateway IP address:

```
rommon> GATEWAY=ip_address
```

Step 9 Verify that you have access to the TFTP server by pinging it from your local Ethernet port with one of the following commands:

```
rommon> ping server_ip_address
rommon> ping server
```

Step 10 If necessary define the path and filename on the TFTP file server from which you are downloading the image:

```
rommon> IMAGE=path/file_name
```

Caution Make sure that you enter the **IMAGE** command in all uppercase. You can enter the other ROMMON commands in either lower case or upper case, but the **IMAGE** command specifically must be all uppercase.

UNIX example:

```
rommon> IMAGE=/system_images/IPS-4240-K9-sys-1.1-a-5.1-5-E1.img
```

Note The path is relative to the UNIX TFTP server's default tftpboot directory. Images located in the default tftpboot directory do not have any directory names or slashes in the IMAGE specification.

Windows example:

```
rommon> IMAGE=C:\system_images\IPS-4240-K9-sys-1.1-a-5.1-5-E1.img
```

Step 11 Type **set** and press **Enter** to verify the network settings.

Note You can use the **sync** command to store these settings in NVRAM so they are maintained across boots. Otherwise, you must type this information each time you want to boot an image from ROMMON.

Step 12 Download and install the system image:

```
rommon> tftp
```

Caution To avoid corrupting the system image, do not remove power from IPS-4240 while the system image is being installed.

Note If the network settings are correct, the system downloads and boots the specified image on IPS-4240. Be sure to use the IPS-4240 image.

Installing the IPS-4260 System Image

You can install the IPS-4260 system image by using the ROMMON on the appliance to TFTP the system image onto the flash device.

To install the IPS-4260 system image, follow these steps:

Step 1 Download the IPS-4260 system image file (IPS-4260-K9-sys-1.1-a-5.1-5-E1.img) to the tftp root directory of a TFTP server that is accessible from your IPS-4260. For the procedure for locating software on Cisco.com, see Obtaining Cisco IPS Software, page 18-1. Make sure you can access the TFTP server location from the network connected to your IPS-4260 Ethernet port.

Step 2 Boot IPS-4260.

Step 3 Press **Ctrl-R** at the following prompt while the system is booting:

```
Evaluating Run Options...
```

Note You have five seconds to press **Ctrl-R**.

The console display resembles the following:

```
Assuming IPS-4260-K9 Platform
 2 Ethernet Interfaces detected

Cisco Systems ROMMON Version (1.0(11)1c) #26: Mon Mar 13 18:05:54 CST 2006

Platform IPS-4260-K9
Management0/0
Link is UP
MAC Address: 0004.23cc.6047


Use ? for help.
rommon #0>
```

Step 4 If necessary, change the port used for the TFTP download:

```
rommon #1> interface name
```

The port in use is listed just after the platform identification. In the example, port Management0/0 is being used.

Note The default port used for TFTP downloads is Management0/0, which corresponds with the command and control (MGMT) interface of the IPS-4260.

Note Ports Management0/0 (MGMT) and GigabitEthernet0/1 (GE 0/1) are labeled on the back of the chassis.

Step 5 Specify an IP address for the local port on IPS-4260:

```
rommon> address ip_address
```

Note Use the same IP address that is assigned to IPS-4260.

Step 6 Specify the TFTP server IP address:

```
rommon> server ip_address
```

Step 7 Specify the gateway IP address:

```
rommon> gateway ip_address
```

Step 8 Verify that you have access to the TFTP server by pinging it from the local Ethernet port:

```
rommon> ping server_ip_address
rommon> ping server
```

Step 9 Specify the path and filename on the TFTP file server from which you are downloading the image:

```
rommon> file path/filename
```

UNIX example:

```
rommon> file /system_images/IPS-4260-K9-sys-1.1-a-5.1-5-E1.img
```

Note The path is relative to the UNIX TFTP server's default tftpboot directory. Images located in the default tftpboot directory do not have any directory names or slashes in the file location.

Windows example:

```
rommon> file <tftpboot_directory>IPS-4260-K9-sys-1.1-a-5.1-5-E1.img
```

Step 10 Download and install the system image:

```
rommon> tftp
```

Note IPS-4260 reboots once during the reimaging process. Do not remove power from IPS-4260 during the update process or the upgrade can become corrupted.

Using the Recovery/Upgrade CD

You can use the recovery/upgrade CD on appliances that have a CD-ROM, such as the IDS-4210, IDS-4235, and IDS-4250. The recovery/upgrade CD reimages both the recovery and application partitions.

Caution You are installing a new software image. All configuration data is overwritten.

After you install the system image with the recovery/upgrade CD, you must use the **setup** command to initialize the appliance. You will need your configuration information. You can obtain this information by generating a diagnostics report through IDM.

Signature updates occur approximately every week or more often if needed. The most recent signature update will not be on the recovery/upgrade CD that shipped with your appliance. Download the most recent signature update and apply it after you have recovered the system image.

To recover the system image with the recovery/upgrade CD, follow these steps:

Step 1 Obtain your configuration information from IDM:

a. To access IDM, point your browser to the appliance you are upgrading.

b. Select **Monitoring > Diagnostics**. The Diagnostics panel appears.

c. Click **Run Diagnostics**. Running the diagnostics may take a while.

d. Click **View Results**. The results are displayed in a report.

e. To save the diagnostics report, select **Menu > Save As** in your browser.

Step 2 Insert the recovery/upgrade CD into the CD-ROM drive.

Step 3 Power off the appliance and then power it back on.

The boot menu appears, which lists important notices and boot options.

Step 4 Type **k** if you are installing from a keyboard, or type **s** if you are installing from a serial connection.

Note A blue screen is displayed for several minutes without any status messages while the files are being copied from the CD to your appliance.

Step 5 Log in to the appliance by using a serial connection or with a monitor and keyboard.

Note The default username and password are both cisco.

Step 6 You are prompted to change the default password.

Note Passwords must be at least eight characters long and be strong, that is, not be a dictionary word.

After you change the password, the `sensor#` prompt appears.

Step 7 Type the **setup** command to initialize the appliance. For the procedure, see Initializing the Sensor, page 3-2.

Step 8 Install the most recent service pack and signature update. For the procedure for locating software on Cisco.com, see Obtaining Cisco IPS Software, page 18-1.

Installing the NM-CIDS System Image

This section describes how to install the NM-CIDS system image, and contains the following topics:

Overview

You can reimage the NM-CIDS using the system image file (IPS-NM-CIDS-K9-sys-1.1-a-5.1-1.pkg). If NM-CIDS is already running version 5.0, the bootloader has been upgraded. If NM-CIDS is not running 5.0, you must upgrade the bootloader before installing the 5.1 image. For the procedure to upgrade the bootloader, refer to Installing the NM-CIDS System Image.

Installing the NM-CIDS System Image

Note The bootloader has a timeout of 10 minutes, which means reimages over slow WAN links will fail. To avoid this situation, download the bootloader file to a local TFTP server and have the NM-CIDS reimage from the local TFTP server.

To reimage NM-CIDS, follow these steps:

Step 1 Download the NM-CIDS system image file (IPS-NM-CIDS-K9-sys-1.1-a-5.1-5-E1.img) to the TFTP root directory of a TFTP server that is accessible from your NM-CIDS. For the procedure for locating software on Cisco.com, see Obtaining Cisco IPS Software, page 18-1.

Note Make sure you can access the TFTP server location from the network connected to the NM-CIDS Ethernet port.

Step 2 Log in to the router.

Step 3 Enter enable mode:

```
router# enable
router(enable)#
```

Step 4 Session to NM-CIDS:

```
router(enable)# service-module IDS-Sensor slot_number/0 session
```

Note Use the **show configuration | include interface IDS-Sensor** command to determine the NM-CIDS slot number.

Step 5 Suspend the session by pressing **Shift-Ctrl-6 X**. You should see the `router#` prompt. If you do not see this prompt, try **Ctrl-6 X**.

Step 6 Reset NM-CIDS:

```
router(enable)# service-module IDS-Sensor slot_number/0 reset
```

You are prompted to confirm the **reset** command.

Step 7 Press **Enter** to confirm.

Step 8 Press **Enter** to resume the suspended session. After displaying its version, the bootloader displays this prompt for 15 seconds:

```
Please enter '***' to change boot configuration:
```

Step 9 Enter ******* during the 15-second delay. The bootloader prompt appears.

Step 10 Display the bootloader configuration:

```
ServicesEngine boot-loader> show config
```

Caution If the bootloader version is not 1.0.17-1, you must upgrade it before installing 5.1. For the procedure, refer to Installing the NM-CIDS System Image.

Step 11 Configure the bootloader parameters:

```
ServicesEngine boot-loader> config
```

Step 12 You are prompted for each value line by line.

a. Specify the IP address—The external fast Ethernet port on NM-CIDS. This must be a real IP address on your network.

b. Specify the subnet mask—The external fast Ethernet port on NM-CIDS. This must be a real IP address on your network.

c. Specify the TFTP server IP address—The IP address of the TFTP server from which to download the NM-CIDS system image.

d. Specify the gateway IP address—The IP address of the default gateway for hosts on your subnet.

e. Specify the default helper file—The name of the helper image to boot. The NM-CIDS helper file is boot helper IPS-NM-CIDS-K9-sys-1.1-a-5.1-1.img.

f. Specify the Ethernet interface—The Ethernet interface is always set to **external**.

g. Specify the default boot device—The default boot device is always set to **disk**.

h. Specify the default bootloader—The default bootloader is always set to **primary**.

If you made any changes, the bootloader stores them permanently. The bootloader command prompt appears.

Caution The next step erases all data from the NM-CIDS hard-disk drive.

Step 13 Boot the system image:

```
ServicesEngine boot-loader> boot helper IPS-NM-CIDS-K9-sys-1.1-a-5.1-5-E1.img
```

The bootloader displays a spinning line while loading the system image from the TFTP server. When the system image is loaded, it is booted. The system image installs IPS 5.1(1) on NM-CIDS. When the installation is complete, NM-CIDS reboots. The system is restored to default settings. The user account and password are set to `cisco`.

You must initialize NM-CIDS with the **setup** command. For the procedure, see Initializing the Sensor, page 3-2.

Installing the IDSM-2 System Image

If the IDSM-2 application partition becomes unusable, you can reimage it from the maintenance partition. After you reimage the application partition of IDSM-2, you must initialize IDSM-2 using the **setup** command. For the procedure, see Initializing the Sensor, page 3-2.

When there is a new maintenance partition image file, you can reimage the maintenance partition from the application partition.

This section describes how to reimage the application partition and maintenance partition for Catalyst software and Cisco IOS software. It contains the following topics:

Installing the System Image

This section describes how to install the IDSM-2 system image, and contains the following topics:

Catalyst Software

To install the system image, follow these steps:

Step 1 Download the IDSM-2 system image file (WS-SVC-IDSM2-K9-sys-1.1-a-5.1-5-E1.bin.gz) to the FTP root directory of a FTP server that is accessible from your IDSM-2. For the procedure for locating software on Cisco.com, see Obtaining Cisco IPS Software, page 18-1.

Step 2 Log in to the switch CLI.

Step 3 Boot IDSM-2 to the maintenance partition:

```
cat6k> (enable) reset module_number cf:1
```

Step 4 Log in to the maintenance partition CLI:

```
login: guest
Password: cisco
```

Note You must configure the maintenance partition on IDSM-2. For the procedure, see Configuring the Maintenance Partition, page 17-29.

Step 5 Install the system image:

```
guest@hostname.localdomain# upgrade ftp://user@ftp server IP/directory
path/WS-SVC-IDSM2-K9-sys-1.1-a-5.1-5-E1.bin.gz
```

Step 6 Specify the FTP server password. After the application partition file has been downloaded, you are asked if you want to proceed:

```
Upgrading will wipe out the contents on the hard disk. Do you want to proceed installing
it [y|n]:
```

Step 7 Type **y** to continue. When the application partition file has been installed, you are returned to the maintenance partition CLI.

Step 8 Exit the maintenance partition CLI and return to the switch CLI.

Step 9 Reboot IDSM-2 to the application partition:

```
cat6k> (enable) reset module_number hdd:1
```

Step 10 When IDSM-2 has rebooted, check the software version. For the procedure, see Verifying IDSM-2 Installation, page 15-2.

Step 11 Log in to the application partition CLI and initialize IDSM-2. For the procedure, see Initializing the Sensor, page 3-2.

Cisco IOS Software

To install the system image, follow these steps:

Step 1 Download the IDSM-2 system image file (WS-SVC-IDSM2-K9-sys-1.1-a-5.1-5-E1.bin.gz) to the FTP root directory of an FTP server that is accessible from your IDSM-2.

Step 2 Log in to the switch CLI.

Step 3 Boot IDSM-2 to the maintenance partition.

```
router# hw-module module module_number reset cf:1
```

Step 4 Session to the maintenance partition CLI.

```
router# session slot slot_number processor 1
```

Step 5 Log in to the maintenance partition CLI.

```
login: guest
Password: cisco
```

Step 6 Configure the maintenance partition interface IP address.

```
guest@localhost.localdomain# ip address ip_address netmask
```

Note Choose an address that is appropriate for the VLAN on which the IDSM-2 management interface is located based on the switch configuration.

Step 7 Configure the maintenance partition default gateway address.

```
guest@localhost.localdomain# ip gateway gateway_address
```

Step 8 Install the system image.

```
guest@hostname.localdomain# upgrade
ftp://user@ftp_server_ip_address/directory_path/WS-SVC-IDSM2-K9-sys-1.1-a-5.1-5-E1.bin.gz
-install
```

Step 9 Specify the FTP server password. After the application partition file has been downloaded, you are asked if you want to proceed:

```
Upgrading will wipe out the contents on the hard disk.
Do you want to proceed installing it [y|n]:
```

Step 10 Enter **y** to continue. When the application partition file has been installed, you are returned to the maintenance partition CLI.

Step 11 Exit the maintenance partition CLI and return to the switch CLI.

Step 12 Reboot IDSM-2 to the application partition.

```
router# hw-module module module_number reset hdd:1
```

Step 13 Verify that IDSM-2 is online and that the software version is correct and that the status is `ok`.

```
router# show module module_number
```

Step 14 Session to the IDSM-2 application partition CLI.

```
router# session slot slot_number processor 1
```

Step 15 Initialize IDSM-2 using the **setup** command. For the procedure, see Initializing the Sensor, page 3-2.

Configuring the Maintenance Partition

This section describes how to configure the maintenance partition on IDSM-2, and contains the following topics:

Catalyst Software

To configure the IDSM-2 maintenance partition, follow these steps:

Step 1 Log in to the switch CLI.

Step 2 Enter privileged mode:

```
cat6k# enable
cat6k(enable)#
```

Step 3 Session to IDSM-2:

```
cat6k# session 9
Trying IDS-9...
Connected to IDS-9.
Escape character is '^]'.

Cisco Maintenance image
```

Note You cannot Telnet or SSH to the IDSM-2 maintenance partition.You must session to it from the switch CLI.

Step 4 Log in as user **guest** and password **cisco**.

Note You can change the guest password, but we do not recommend it. If you forget the maintenance partition guest password, and you cannot log in to the IDSM-2 application partition for some reason, you will have to RMA IDSM-2.

```
login: guest
Password: cisco

Maintenance image version: 2.1(2)

guest@idsm2.localdomain#
```

Step 5 View the IDSM-2 maintenance partition host configuration:

```
guest@idsm2.localdomain# show ip

IP address      : 10.89.149.74
Subnet Mask     : 255.255.255.128
IP Broadcast    : 10.255.255.255
DNS Name        : idsm2.localdomain
Default Gateway : 10.89.149.126
Nameserver(s)   :

guest@idsm2.localdomain#
```

Step 6 Clear the IDSM-2 maintenance partition host configuration (ip address, gateway, hostname):

```
guest@idsm2.localdomain# clear ip
guest@localhost.localdomain# show ip

IP address      : 0.0.0.0
Subnet Mask     : 0.0.0.0
IP Broadcast    : 0.0.0.0
DNS Name        : localhost.localdomain
Default Gateway : 0.0.0.0
Nameserver(s)   :

guest@localhost.localdomain#
```

Step 7 Configure the maintenance partition host configuration:

a. Specify the IP address:

```
guest@localhost.localdomain# ip address ip_address netmask
```

b. Specify the default gateway:

```
guest@localhost.localdomain# ip gateway gateway_ip_address
```

c. Specify the hostname:

```
guest@localhost.localdomain# ip host hostname
```

Step 8 View the maintenance partition host configuration:

```
guest@idsm2.localdomain# show ip

IP address      : 10.89.149.74
Subnet Mask     : 255.255.255.128
IP Broadcast    : 10.255.255.255
DNS Name        : idsm2.localdomain
Default Gateway : 10.89.149.126
Nameserver(s)   :

guest@idsm2.localdomain#
```

Step 9 Verify the image installed on the application partition:

```
guest@idsm2.localdomain# show images
Device name             Partition#              Image name
```

```
-----------             ----------              ----------
Hard disk(hdd)          1                       5.0(1)
guest@idsm2.localdomain#
```

Step 10 Verify the maintenance partition version (including the BIOS version):

```
guest@idsm2.localdomain# show version

Maintenance image version: 2.1(2)
mp.2-1-2.bin : Thu Nov 18 11:41:36 PST 2004 :
integ@kplus-build-lx.cisco.com

Line Card Number :WS-SVC-IDSM2-XL
Number of Pentium-class Processors : 2
BIOS Vendor: Phoenix Technologies Ltd.
BIOS Version: 4.0-Rel 6.0.9

Total available memory: 2012 MB
Size of compact flash: 61 MB
Size of hard disk: 19077 MB
Daughter Card Info: Falcon rev 3, FW ver 2.0.3.0 (IDS), SRAM  8 MB, SDRAM  256 MB

guest@idsm2.localdomain#
```

Step 11 Upgrade the application partition:

```
guest@idsm2.localdomain# upgrade
ftp://jsmith@10.89.146.11//RELEASES/Latest/5.0-1/WS-SVC-IDSM2-K9-sys-1.1-a-5.0-1.bin.gz
Downloading the image. This may take several minutes...
Password for jsmith@10.89.146.114:
500 'SIZE WS-SVC-IDSM2-K9-sys-1.1-a-5.0-1.bin.gz': command not understood.

ftp://jsmith@10.89.146.11//RELEASES/Latest/5.0-1/WS-SVC-IDSM2-K9-sys-1.1-a-5.0-1.bin.gz
(unknown size)
/tmp/upgrade.gz           [|]   28616K
29303086 bytes transferred in 5.34 sec (5359.02k/sec)

Upgrade file
ftp://jsmith@10.89.146.114//RELEASES/Latest/5.0-1/WS-SVC-IDSM2-K9-sys-1.1-a-5.0-1.bin.gz
is downloaded.
Upgrading will wipe out the contents on the storage media.
Do you want to proceed installing it [y|N]:
```

Step 12 Type **y** to proceed with the upgrade.

```
Proceeding with upgrade. Please do not interrupt.
If the upgrade is interrupted or fails, boot into maintenance image again and restart
upgrade.

Creating IDS application image file...

Initializing the hard disk...
Applying the image, this process may take several minutes...
Performing post install, please wait...
Application image upgrade complete. You can boot the image now.
guest@idsm3.localdomain#
```

Step 13 Display the upgrade log:

```
guest@idsm3.localdomain# show log upgrade

Upgrading the line card on Fri Mar 11 21:21:53 UTC 2005
Downloaded upgrade image
ftp://jsmith@10.89.146.114//RELEASES/Latest/5.0-1/WS-SVC-IDSM2-K9-sys-1.1-a-5.0-1.bin.gz
Extracted the downloaded file
```

```
Proceeding with image upgrade.
Fri Mar 11 21:22:06 2005 : argv1 = 0, argv2 = 0, argv3 = 3, argv4 = 1
Fri Mar 11 21:22:06 2005 : Creating IDS application image file...
Fri Mar 11 21:22:06 2005 : footer: XXXXXXXXXXXXXXXX
Fri Mar 11 21:22:06 2005 : exeoff: 0000000000031729
Fri Mar 11 21:22:06 2005 : image: 0000000029323770
Fri Mar 11 21:22:06 2005 : T: 29323818, E: 31729, I: 29323770
Fri Mar 11 21:22:07 2005 : partition: /dev/hdc1
Fri Mar 11 21:22:07 2005 : startIDSAppUpgrade:Image: /tmp/cdisk.gz
Fri Mar 11 21:22:07 2005 : startIDSAppUpgrade:Device: /dev/hdc1
Fri Mar 11 21:22:07 2005 : startIDSAppUpgrade:Install type: 1
Fri Mar 11 21:22:07 2005 : Initializing the hard disk...
Fri Mar 11 21:22:07 2005 : Required disk size: 524288 Kb (blocks)
Fri Mar 11 21:22:07 2005 : Available disk size: 19535040 Kb (blocks)
Fri Mar 11 21:22:13 2005 : Partitions created on '/dev/hdc'.
Fri Mar 11 21:22:13 2005 : Device '/dev/hdc' verified for OK.
Fri Mar 11 21:22:19 2005 : Created ext2 fileSystem on '/dev/hdc1'.
Fri Mar 11 21:22:19 2005 : Directory '/mnt/hd/' created.
Fri Mar 11 21:22:19 2005 : Partition '/dev/hdc1' mounted.
Fri Mar 11 21:22:19 2005 : Finished initializing the hard disk.
Fri Mar 11 21:22:19 2005 : Applying the image, this process may take several minutes...
Fri Mar 11 21:22:19 2005 : Directory changed to '/mnt/hd'.
Fri Mar 11 21:22:20 2005 : Performing post install, please wait...
Fri Mar 11 21:22:20 2005 : File /mnt/hd/post-install copied to /tmp/post-install.
Fri Mar 11 21:22:20 2005 : Directory changed to '/tmp'.
Fri Mar 11 21:22:28 2005 : Partition '/dev/hdc1' unmounted.
Fri Mar 11 21:22:28 2005 : Directory changed to '/tmp'.
Application image upgrade complete. You can boot the image now.
Partition upgraded successfully
guest@idsm2.localdomain#
```

Step 14 Clear the upgrade log:

```
guest@idsm2.localdomain# clear log upgrade
Cleared log file successfully
```

Step 15 Display the upgrade log:

```
guest@idsm2.localdomain# show log upgrade
guest@idsm2.localdomain#
```

Step 16 Ping another computer:

```
guest@idsm2.localdomain# ping 10.89.146.114
PING 10.89.146.114 (10.89.146.114) from 10.89.149.74 : 56(84) bytes of data.
64 bytes from 10.89.146.114: icmp_seq=0 ttl=254 time=381 usec
64 bytes from 10.89.146.114: icmp_seq=1 ttl=254 time=133 usec
64 bytes from 10.89.146.114: icmp_seq=2 ttl=254 time=129 usec
64 bytes from 10.89.146.114: icmp_seq=3 ttl=254 time=141 usec
64 bytes from 10.89.146.114: icmp_seq=4 ttl=254 time=127 usec

--- 10.89.146.114 ping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max/mdev = 0.127/0.182/0.381/0.099 ms
guest@idsm2.localdomain#
```

Step 17 Reset IDSM-2:

Note You cannot specify a partition when issuing the **reset** command from the maintenance partition. IDSM-2 boots to whichever partition is specified in the boot device variable. If the boot device variable is blank, IDSM-2 boots to the application partition.

```
guest@idsm2.localdomain# reset
guest@idsm2.localdomain#
2005 Mar 11 21:55:46 CST -06:00 %SYS-4-MOD_SHUTDOWNSTART:Module 9 shutdown in progress. Do
not remove module until shutdown completes

Broadcast message from root Fri Mar 11 21:55:47 2005...

The system is going down for system halt NOW !!
cat6k> (enable)
```

Cisco IOS Software

To configure the IDSM-2 maintenance partition, follow these steps:

Step 1 Log in to the switch CLI.

Step 2 Session to IDSM-2:

```
switch# session slot 11 processor 1
The default escape character is Ctrl-^, then x.
You can also type 'exit' at the remote prompt to end the session
Trying 127.0.0.111 ... Open

Cisco Maintenance image
```

Note You cannot Telnet or SSH to the IDSM-2 maintenance partition.You must session to it from the switch CLI.

Step 3 Log in as user **guest** and password **cisco**.

Note You can change the guest password, but we do not recommend it. If you forget the maintenance partition guest password, and you cannot log in to the IDSM-2 application partition for some reason, you will have to RMA IDSM-2.

```
login: guest
password: cisco

Maintenance image version: 2.1(2)

guest@idsm2.localdomain#
```

Step 4 View the maintenance partition host configuration:

```
guest@idsm2.localdomain# show ip

IP address      : 10.89.149.74
Subnet Mask     : 255.255.255.128
IP Broadcast    : 10.255.255.255
DNS Name        : idsm2.localdomain
Default Gateway : 10.89.149.126
Nameserver(s)   :

guest@idsm2.localdomain#
```

Step 5 Clear the maintenance partition host configuration (ip address, gateway, hostname):

```
guest@idsm2.localdomain# clear ip
guest@localhost.localdomain# show ip

IP address      : 0.0.0.0
Subnet Mask     : 0.0.0.0
IP Broadcast    : 0.0.0.0
DNS Name        : localhost.localdomain
Default Gateway : 0.0.0.0
Nameserver(s)   :

guest@localhost.localdomain#
```

Step 6 Configure the maintenance partition host configuration:

a. Specify the IP address:

```
guest@localhost.localdomain# ip address ip_address netmask
```

b. Specify the default gateway:

```
guest@localhost.localdomain# ip gateway gateway_ip_address
```

c. Specify the hostname:

```
guest@localhost.localdomain# ip host hostname
```

Step 7 View the maintenance partition host configuration:

```
guest@idsm2.localdomain# show ip

IP address      : 10.89.149.74
Subnet Mask     : 255.255.255.128
IP Broadcast    : 10.255.255.255
DNS Name        : idsm2.localdomain
Default Gateway : 10.89.149.126
Nameserver(s)   :

guest@idsm2.localdomain#
```

Step 8 Verify the image installed on the application partition:

```
guest@idsm2.localdomain# show images
Device name             Partition#              Image name
-----------             ----------               ----------
Hard disk(hdd)          1                       5.0(1)
guest@idsm2.localdomain#
```

Step 9 Verify the maintenance partition version (including the BIOS version):

```
guest@idsm2.localdomain# show version

Maintenance image version: 2.1(2)
mp.2-1-2.bin : Thu Nov 18 11:41:36 PST 2004 :
integ@kplus-build-lx.cisco.com

Line Card Number :WS-SVC-IDSM2-XL
Number of Pentium-class Processors : 2
BIOS Vendor: Phoenix Technologies Ltd.
BIOS Version: 4.0-Rel 6.0.9

Total available memory: 2012 MB
Size of compact flash: 61 MB
Size of hard disk: 19077 MB
```

```
Daughter Card Info: Falcon rev 3, FW ver 2.0.3.0 (IDS), SRAM  8 MB, SDRAM  256 MB

guest@idsm2.localdomain#
```

Step 10 Upgrade the application partition:

```
guest@idsm2.localdomain# upgrade
ftp://jsmith@10.89.146.11//RELEASES/Latest/5.0-1/WS-SVC-IDSM2-K9-sys-1.1-a-5.0-1.bin.gz
Downloading the image. This may take several minutes...
Password for jsmith@10.89.146.114:
500 'SIZE WS-SVC-IDSM2-K9-sys-1.1-a-5.0-1.bin.gz': command not understood.

ftp://jsmith@10.89.146.11//RELEASES/Latest/5.0-1/WS-SVC-IDSM2-K9-sys-1.1-a-5.0-1.bin.gz
(unknown size)
/tmp/upgrade.gz           [|]   28616K
29303086 bytes transferred in 5.34 sec (5359.02k/sec)

Upgrade file
ftp://jsmith@10.89.146.114//RELEASES/Latest/5.0-1/WS-SVC-IDSM2-K9-sys-1.1-a-5.0-1.bin.gz
is downloaded.
Upgrading will wipe out the contents on the storage media.
Do you want to proceed installing it [y|N]:
```

Step 11 Type **y** to proceed with the upgrade.

```
Proceeding with upgrade. Please do not interrupt.
If the upgrade is interrupted or fails, boot into maintenance image again and restart
upgrade.

Creating IDS application image file...

Initializing the hard disk...
Applying the image, this process may take several minutes...
Performing post install, please wait...
Application image upgrade complete. You can boot the image now.
guest@idsm3.localdomain#
```

Step 12 Display the upgrade log:

```
guest@idsm3.localdomain# show log upgrade

Upgrading the line card on Fri Mar 11 21:21:53 UTC 2005
Downloaded upgrade image
ftp://jsmith@10.89.146.114//RELEASES/Latest/5.0-1/WS-SVC-IDSM2-K9-sys-1.1-a-5.0-1.bin.gz
Extracted the downloaded file
Proceeding with image upgrade.
Fri Mar 11 21:22:06 2005 : argv1 = 0, argv2 = 0, argv3 = 3, argv4 = 1
Fri Mar 11 21:22:06 2005 : Creating IDS application image file...
Fri Mar 11 21:22:06 2005 : footer: XXXXXXXXXXXXXXXX
Fri Mar 11 21:22:06 2005 : exeoff: 0000000000031729
Fri Mar 11 21:22:06 2005 : image: 0000000029323770
Fri Mar 11 21:22:06 2005 : T: 29323818, E: 31729, I: 29323770
Fri Mar 11 21:22:07 2005 : partition: /dev/hdc1
Fri Mar 11 21:22:07 2005 : startIDSAppUpgrade:Image: /tmp/cdisk.gz
Fri Mar 11 21:22:07 2005 : startIDSAppUpgrade:Device: /dev/hdc1
Fri Mar 11 21:22:07 2005 : startIDSAppUpgrade:Install type: 1
Fri Mar 11 21:22:07 2005 : Initializing the hard disk...
Fri Mar 11 21:22:07 2005 : Required disk size: 524288 Kb (blocks)
Fri Mar 11 21:22:07 2005 : Available disk size: 19535040 Kb (blocks)
Fri Mar 11 21:22:13 2005 : Partitions created on '/dev/hdc'.
Fri Mar 11 21:22:13 2005 : Device '/dev/hdc' verified for OK.
Fri Mar 11 21:22:19 2005 : Created ext2 fileSystem on '/dev/hdc1'.
Fri Mar 11 21:22:19 2005 : Directory '/mnt/hd/' created.
Fri Mar 11 21:22:19 2005 : Partition '/dev/hdc1' mounted.
Fri Mar 11 21:22:19 2005 : Finished initializing the hard disk.
```

```
Fri Mar 11 21:22:19 2005 : Applying the image, this process may take several minutes...
Fri Mar 11 21:22:19 2005 : Directory changed to '/mnt/hd'.
Fri Mar 11 21:22:20 2005 : Performing post install, please wait...
Fri Mar 11 21:22:20 2005 : File /mnt/hd/post-install copied to /tmp/post-install.
Fri Mar 11 21:22:20 2005 : Directory changed to '/tmp'.
Fri Mar 11 21:22:28 2005 : Partition '/dev/hdc1' unmounted.
Fri Mar 11 21:22:28 2005 : Directory changed to '/tmp'.
Application image upgrade complete. You can boot the image now.
Partition upgraded successfully
guest@idsm2.localdomain#
```

Step 13 Clear the upgrade log:

```
guest@idsm2.localdomain# clear log upgrade
Cleared log file successfully
```

Step 14 Display the upgrade log:

```
guest@idsm2.localdomain# show log upgrade
guest@idsm2.localdomain#
```

Step 15 Ping another computer:

```
guest@idsm2.localdomain# ping 10.89.146.114
PING 10.89.146.114 (10.89.146.114) from 10.89.149.74 : 56(84) bytes of data.
64 bytes from 10.89.146.114: icmp_seq=0 ttl=254 time=381 usec
64 bytes from 10.89.146.114: icmp_seq=1 ttl=254 time=133 usec
64 bytes from 10.89.146.114: icmp_seq=2 ttl=254 time=129 usec
64 bytes from 10.89.146.114: icmp_seq=3 ttl=254 time=141 usec
64 bytes from 10.89.146.114: icmp_seq=4 ttl=254 time=127 usec

--- 10.89.146.114 ping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max/mdev = 0.127/0.182/0.381/0.099 ms
guest@idsm2.localdomain#
```

Step 16 Reset IDSM-2:

Note You cannot specify a partition when issuing the **reset** command from the maintenance partition. IDSM-2 boots to whichever partition is specified in the boot device variable. If the boot device variable is blank, IDSM-2 boots to the application partition.

```
guest@idsm2.localdomain# reset
guest@idsm2.localdomain#
Broadcast message from root Fri Mar 11 22:04:53 2005...

The system is going down for system halt NOW !!

[Connection to 127.0.0.111 closed by foreign host]
switch#
```

Upgrading the Maintenance Partition

This section describes how to upgrade the maintenance partition, and contains the following topics:

- Catalyst Software, page 17-37
- Cisco IOS Software, page 17-37

Catalyst Software

To upgrade the maintenance partition, follow these steps:

Step 1 Download the IDSM-2 maintenance partition file (c6svc-mp.2-1-1.bin.gz) to the FTP root directory of a FTP server that is accessible from your IDSM-2. For the procedure for locating software on Cisco.com, see Obtaining Cisco IPS Software, page 18-1.

Step 2 Log in to the IDSM-2 CLI.

Step 3 Enter configuration mode:

```
idsm2# configure terminal
```

Step 4 Upgrade the maintenance partition:

```
idsm2# upgrade ftp://user@ftp_server_IP_address/directory_path/c6svc-mp.2-1-1.bin.gz
```

You are asked whether you want continue.

Step 5 Type **y** to continue. The maintenance partition file is upgraded.

Cisco IOS Software

To upgrade the maintenance partition, follow these steps:

Step 1 Download the IDSM-2 maintenance partition file (c6svc-mp.2-1-1.bin.gz) to the FTP root directory of a FTP server that is accessible from your IDSM-2. For the procedure for locating software on Cisco.com, see Obtaining Cisco IPS Software, page 18-1.

Step 2 Log in to the switch CLI.

Step 3 Session in to the application partition CLI:

```
switch# session slot slot_number processor 1
```

Step 4 Enter configuration mode:

```
idsm2# configure terminal
```

Step 5 Upgrade the maintenance partition:

```
idsm2(config)# upgrade
ftp://user@ftp_server_IP_address/directory_path/c6svc-mp.2-1-1.bin.gz
```

Step 6 Specify the FTP server password:

```
Password: ********
```

You are prompted to continue:

```
Continue with upgrade?:
```

Step 7 Type **yes** to continue.

Installing the AIP-SSM System Image

You can reimage the AIP-SSM in one of the following ways:

- From ASA using the **hw-module module 1 recover configure/boot** command. See the following procedure.
- Recovering the application image from the sensor's CLI using the **recover application-partition** command. For the procedure, see Recovering the Application Partition, page 17-11.
- Upgrading the recovery image from the sensor's CLI using the **upgrade** command. For the procedure, see Upgrading the Recovery Partition, page 17-5.

To install the AIP-SSM system image, follow these steps:

Step 1 Download the AIP-SSM system image file (IPS-SSM-K9-sys-1.1-a-5.1-5-E1.img) to the TFTP root directory of a TFTP server that is accessible from your AIP-SSM. For the procedure for locating software on Cisco.com, see Obtaining Cisco IPS Software, page 18-1.

Note Make sure you can access the TFTP server location from the network connected to the AIP-SSM Ethernet port.

Step 2 Log in to the ASA.

Step 3 Enter enable mode:

```
asa> enable
```

Step 4 Configure the recovery settings for AIP-SSM:

```
asa# hw-module module 1 recover configure
```

Note If you make an error in the recovery configuration, use the **hw-module module 1 recover stop** command to stop the system reimaging and then you can correct the configuration.

Step 5 Specify the TFTP URL for the system image:

```
Image URL [tftp://0.0.0.0/]:
```

Example:

```
Image URL [tftp://0.0.0.0/]: tftp://10.89.146.1/IPS-SSM-K9-sys-1.1-a-5.1-5-E1.img
```

Step 6 Specify the command and control interface of AIP-SSM:

```
Port IP Address [0.0.0.0]:
```

Example:

```
Port IP Address [0.0.0.0]: 10.89.149.231
```

Step 7 Leave the VLAN ID at 0.

```
VLAN ID [0]:
```

Step 8 Specify the default gateway of the AIP-SSM:

```
Gateway IP Address [0.0.0.0]:
```

Example:

```
Gateway IP Address [0.0.0.0]: 10.89.149.254
```

Step 9 Execute the recovery:

```
asa# hw-module module 1 recover boot
```

Step 10 Periodically check the recovery until it is complete:

Note The status reads `Recovery` during recovery and reads `Up` when reimaging is complete.

```
asa# show module 1

Mod Card Type                                    Model              Serial No.
--- -------------------------------------------- ------------------ -----------
  0 ASA 5540 Adaptive Security Appliance         ASA5540            P2B00000019
  1 ASA 5500 Series Security Services Module-20  ASA-SSM-20         P1D000004F4

Mod MAC Address Range                 Hw Version   Fw Version   Sw Version
--- --------------------------------- ------------ ------------ ---------------
  0 000b.fcf8.7b1c to 000b.fcf8.7b20  0.2          1.0(7)2      7.0(0)82
  1 000b.fcf8.011e to 000b.fcf8.011e  0.1          1.0(7)2      5.0(0.22)S129.0

Mod Status
--- ------------------
  0 Up Sys
  1 Up
asa#
```

Note To debug any errors that may happen in the recovery process, use the **debug module-boot** command to enable debugging of the system reimaging process.

Step 11 Session to AIP-SSM and initialize AIP-SSM with the **setup** command. For the procedure, see Initializing the Sensor, page 3-2.

CHAPTER 18

Obtaining Software

Caution The BIOS on Cisco IDS/IPS sensors is specific to Cisco IDS/IPS sensors and must only be upgraded under instructions from Cisco with BIOS files obtained from the Cisco website. Installing a non-Cisco or third-party BIOS on Cisco IDS/IPS sensors voids the warranty. For more information on how to obtain instructions and BIOS files from the Cisco website, see Obtaining Cisco IPS Software, page 18-1.

This chapter provides information on obtaining Cisco IPS software for the sensor. It contains the following sections:

Obtaining Cisco IPS Software

You can find major and minor updates, service packs, signature and signature engine updates, system and recovery files, firmware upgrades, and readmes on the Download Software site on Cisco.com.

Note You must be logged in to Cisco.com to download software.

Signature updates are posted to Cisco.com approximately every week, more often if needed. Service packs are posted to Cisco.com as needed. Major and minor updates are also posted periodically. Check Cisco.com regularly for the latest IPS software.

Note You must have an active IPS maintenance contract and a Cisco.com password to download software. You must have a license to apply signature updates.

To download software on Cisco.com, follow these steps:

Step 1 Log in to Cisco.com.

Step 2 From the Support drop-down menu, choose **Download Software**.

Step 3 Under Select a Software Product Category, choose **Security Software**.

Step 4 Choose **Intrusion Prevention System (IPS)**.

Step 5 Enter your username and password.

Step 6 In the Download Software window, choose **IPS Appliances > Cisco Intrusion Prevention System** and then click the version you want to download.

Note You must have an IPS subscription service license to download software.

Step 7 Click the type of software file you need.

The available files appear in a list in the right side of the window. You can sort by file name, file size, memory, and release date. And you can access the Release Notes and other product documentation.

Step 8 Click the file you want to download.

The file details appear.

Step 9 Verify that it is the correct file, and click **Download**.

Step 10 Click **Agree** to accept the software download rules.

The first time you download a file from Cisco.com, you must fill in the Encryption Software Export Distribution Authorization form before you can download the software.

- Fill out the form and click **Submit**.

  The Cisco Systems Inc. Encryption Software Usage Handling and Distribution Policy appears.

- Read the policy and click **I Accept**.

  The Encryption Software Export/Distribution Form appears.

If you previously filled out the Encryption Software Export Distribution Authorization form, and read and accepted the Cisco Systems Inc. Encryption Software Usage Handling and Distribution Policy, these forms are not displayed again.

The File Download dialog box appears.

Step 11 Open the file or save it to your computer.

Step 12 Follow the instructions in the Readme to install the update.

Note Major and minor updates, service packs, recovery files, signature and signature engine updates are the same for all sensors. System image files are unique per platform.

IPS Software Versioning

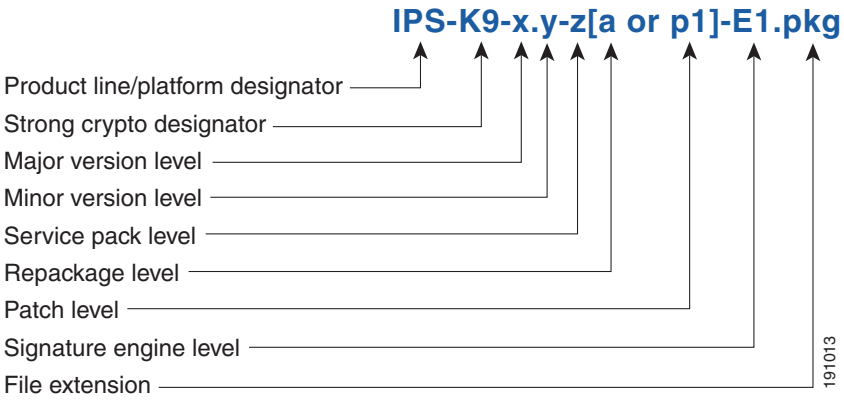
When you download IPS software images from Cisco.com, you should understand the versioning scheme so that you know which files are base files, which are cumulative, and which are incremental

This section describes the various IPS software files, gives software release examples, and contains the following topics:

Major and Minor Updates, Service Packs, and Patch Releases

Figure 18-1 illustrates what each part of the IPS software file represents for major and minor updates, service packs, and patch releases.

Figure 18-1 IPS Software File Name for Major and Minor Updates, Service Packs, and Patch Releases



Major Update

Contains new functionality or an architectural change in the product. For example, the IPS 5.0 base version includes everything (except deprecated features) since the previous major release (the minor update features, service pack fixes, and signature updates) plus any new changes. Major update 5.0(1) requires 4.*x*. With each major update there are corresponding system and recovery packages.

Note The 5.0(1) major update is only used to upgrade 4.*x* sensors to 5.0(1). If you are reinstalling 5.0(1) on a sensor that already has 5.0(1) installed, use the system image or recovery procedures rather than the major update.

Minor Update

Incremental to the major version. Minor updates are also base versions for service packs. The first minor update for 5.0 is 5.1(1). Minor updates are released for minor enhancements to the product. Minor updates contain all previous minor features (except deprecated features), service pack fixes, signature

updates since the last major version, and the new minor features being released. You can install the minor updates on the previous major or minor version (and often even on earlier versions). The minimum supported version needed to upgrade to the newest minor version is listed in the Readme that accompanies the minor update. With each minor update there are corresponding system and recovery packages.

Service Packs

Cumulative following a base version release (minor or major). Service packs are used for the release of defect fixes with no new enhancements. Service packs contain all service pack fixes since the last base version (minor or major) and the new defect fixes being released. Service packs require the minor version. The minimum supported version needed to upgrade to the newest service pack is listed in the Readme that accompanies the service pack. Service packs also include the latest engine update. For example, if service pack 6.0(3) is released, and E3 is the latest engine level, the service pack is released as 6.0(3)E3.

Patch Release

Used to address defects that are identified in the upgrade binaries after a software release. Rather than waiting until the next major or minor update, or service pack to address these defects, a patch can be posted. Patches include all prior patch releases within the associated service pack level. The patches roll into the next official major or minor update, or service pack.

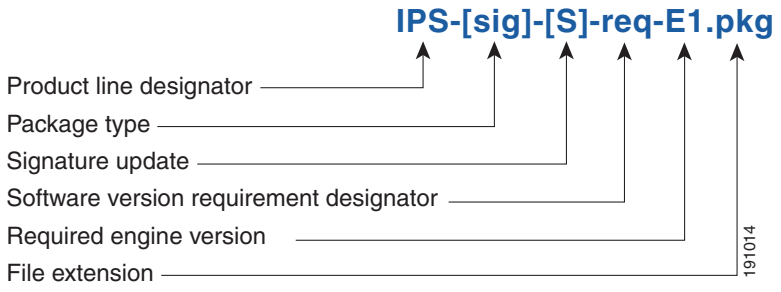
Before you can install a patch release, the most recent major or minor update, or service pack must be installed. For example, patch release 5.0(1p1) requires 5.0(1).

Note Upgrading to a newer patch does not require you to uninstall the old patch. For example, you can upgrade from patch 5.0(1p1) to 5.0(1p2) without first uninstalling 5.0(1p1).

Signature/Virus Updates and Signature Engine Updates

Figure 18-2 illustrates what each part of the IPS software file represents for signature/virus updates.

Figure 18-2 IPS Software File Name for Signature/Virus Updates,



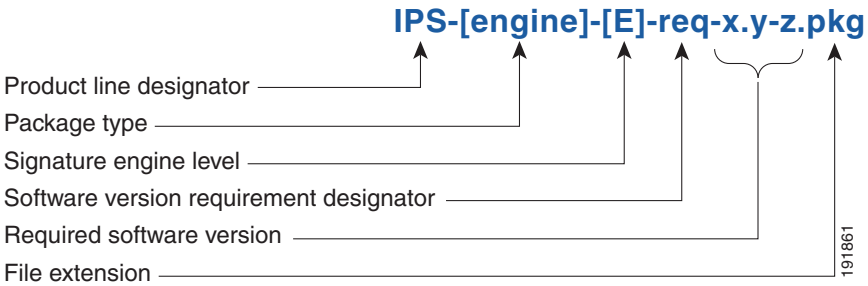
Signature/Virus Updates

Executable file containing a set of rules designed to recognize malicious network activities. Signature updates are released independently from other software updates. Each time a major or minor update is released, you can install signature updates on the new version and the next oldest version for a period of at least six months. Signature updates are dependent on a required signature engine version. Because of this, a *req* designator lists the signature engine required to support a particular signature update.

A virus component for the signature updates is packaged with the signature update. Virus updates are generated by Trend Microsystems for use by the Cisco Intrusion Containment System (Cisco ICS). Once created for use by Cisco ICS, they are later be incorporated into standard Cisco signature updates.

Figure 18-3 illustrates what each part of the IPS software file represents for signature engine updates.

Figure 18-3 IPS Software File Name for Signature Engine Updates



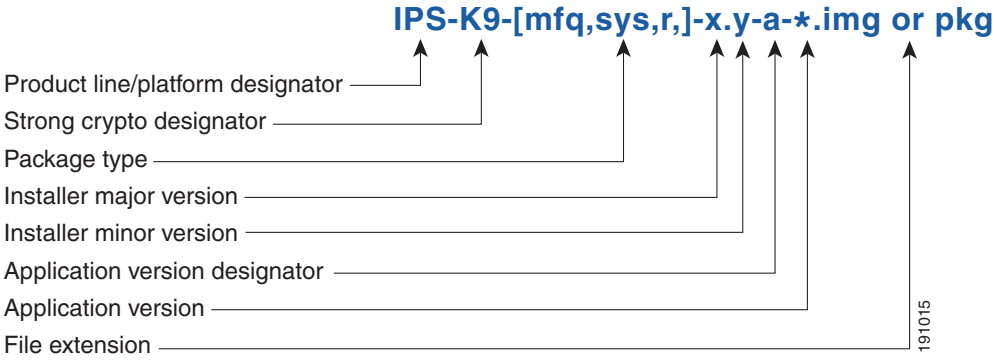
Signature Engine Updates

Executable files containing binary code to support new signature updates. Signature engine files require a specific service pack, which is also identified by the *req* designator.

Recovery, Manufacturing, and System Images

Figure 18-4 illustrates what each part of the IPS software file represents for recovery and system image filenames.

Figure 18-4 IPS Software File Name for Recovery and System Image Filenames



Recovery and system images contain separate versions for the installer and the underlying application. The installer version contains a major and minor version field.

Installer Major Version

The major version is incremented by one of any major changes to the image installer, for example, switching from .tar to rpm or changing kernels.

Installer Minor Version

The minor version can be incremented by any one of the following:

- Minor change to the installer, for example, a user prompt added.
- Repackages require the installer minor version to be incremented by one if the image file must be repackaged to address a defect or problem with the installer.

IPS 5.1 Software Release Examples

Table 18-1 lists platform-independent IDS 5.1(5)E1 software release examples. Refer to the Readmes that accompany the software files for detailed instructions on how to install the files. For instructions on how to access these files on Cisco.com, see Obtaining Cisco IPS Software, page 18-1.

Table 18-1 Platform-Independent Release Examples

Release	Target Frequency	Identifier	Example Version	Example Filename
Signature update[1]	Weekly	sig	S700	IPS-sig-S700-req-E1.pkg
Signature engine update[2]	As needed	engine	E1	IPS-engine-E1-req-5.1-3.pkg
Service packs[3]	Semi-annually or as needed	—	5.1(3)	IPS-K9-5.1-3-E1.pkg
Minor update[4]	Annually	—	5.1(1)	IPS-K9-5.1-1-E1.pkg
Major update[5]	Annually	—	5.0(1)	IPS-K9-6.0-1-E1.pkg
Patch release[6]	As needed	patch	5.0(1p1)	IPS-K9-patch-5.1-1pl-E1.pkg
Recovery package[7]	Annually or as needed	r	1.1-5.0(1)	IPS-K9-r-1.1-a-5.1-1-E1.pkg

1. Signature updates include the latest cumulative IPS signatures.
2. Signature engine updates add new engines or engine parameters that are used by new signatures in later signature updates.
3. Service packs include defect fixes.
4. Minor versions include new minor version features and/or minor version functionality.
5. Major versions include new major version functionality or new architecture.
6. Patch releases are for interim fixes.
7. The r 1.1 can be revised to r 1.2 if it is necessary to release a new recovery package that contains the same underlying application image. If there are defect fixes for the installer, for example, the underlying application version may still be 5.0(1), but the recovery partition image will be r 1.2.

Table 18-2 describes platform-dependent software release examples.

Table 18-2 Platform-Dependent Release Examples

Release	Target Frequency	Identifier	Supported Platform	Example Filename
System image[1]	Annually	sys	Separate file for each sensor platform	IPS-4240-K9-sys-1.1-a-5.1-1-E1.img
Maintenance partition image[2]	Annually	mp	IDSM-2	c5svc-mp.2-1-2.bin.gz
Bootloader	As needed	bl	NM-CIDS AIM-IPS NME-IPS	servicesengine-boot-1.0-4.bin

1. The system image includes the combined recovery and application image used to reimage an entire sensor.
2. The maintenance partition image includes the full image for the IDSM-2 maintenance partition. The file is installed from but does not affect the IDSM-2 application partition.

Table 18-3 describes the platform identifiers used in platform-specific names.

Note IDS-4235 and IDS-4250 do not use platform-specific image files.

Table 18-3 Platform Identifiers

Sensor	Identifier
IDS-4215	IDS-4215-
IPS-4240	IPS-4240-
IPS-4255	IPS-4255-
IPS-4260	IPS-4260-
IDS module for Catalyst 6K	WS-SVC-IDSM2-
IDS network module	IPS-NM-CIDS-
AIP-SSM	IPS-SSM-

Upgrading Cisco IPS Software to 5.0

Note You cannot upgrade the IDSM (WS-X6381) to Cisco IDS 5.*x*. You must replace your IDSM (WS-X6381) with IDSM-2 (WS-SVC-IDSM2-K9), which supports version 5.*x*.

Pay attention to the following when upgrading to IPS 5.0:

- The minimum required version for upgrading to 5.1 is 5.0. The minimum required version for upgrading to 5.0 is 4.1(1). The upgrades from Cisco 5.0 to 5.1 and Cisco 4.1 to 5.0 are available as a downloads from Cisco.com. For the procedure for accessing Downloads on Cisco.com, see Obtaining Cisco IPS Software, page 18-1.
- After downloading the 5.1 upgrade file, refer to the accompanying Readme for the procedure for installing the 5.1 upgrade file using the **upgrade** command. Fro more information, see Upgrading the Sensor, page 17-2.
- After you upgrade any IPS software on your sensor, you must restart the IDM to see the latest software features.
- If you configured Auto Update for your sensor, copy the 5.1 upgrade file to the directory on the server that your sensor polls for updates. See Configuring Automatic Upgrades, page 17-6.
- If you install an upgrade on your sensor and the sensor is unusable after it reboots, you must reimage your sensor. Upgrading a sensor from any Cisco IDS version before 4.1 also requires you to use the **recover** command or the recovery/upgrade CD.

You can reimage your sensor in the following ways:

- For IDS appliances with a CD-ROM drive, use the recovery/upgrade CD.

  For the procedure, see Using the Recovery/Upgrade CD, page 17-23.
- For all sensors, use the **recover** command.

  For the procedure, see Recovering the Application Partition, page 17-11.
- For the IDS-4215, IPS-4240, IPS 4255, and IPS-4260 use the ROMMON to restore the system image.

  For the procedures, see Installing the IDS-4215 System Image, page 17-15, Installing the IPS-4240 and IPS-4255 System Image, page 17-18, and Installing the IPS-4260 System Image, page 17-22.
- For NM-CIDS, use the bootloader.

  For the procedure, see Installing the NM-CIDS System Image, page 17-25.
- For IDSM-2, reimage the application partition from the maintenance partition.

  For the procedure, see Installing the IDSM-2 System Image, page 17-27.
- For AIP-SSM, reimage from ASA using the **hw-module module 1 recover configure/boot** command.

  For the procedure, see Installing the AIP-SSM System Image, page 17-38.

Caution When you install the system image for your sensor, all accounts are removed and the default account and password are reset to cisco.

Obtaining a License Key From Cisco.com

This section describes how to obtain a license key from Cisco.com and how to install it using the CLI or IDM. It contains the following topics:

Overview

Although the sensor functions without the license key, you must have a license key to obtain signature updates. To obtain a license key, you must have the following:

- Cisco Service for IPS service contract

  Contact your reseller, Cisco service or product sales to purchase a contract. For more information, see Service Programs for IPS Products, page 18-10.

- Your IPS device serial number

  To find the IPS device serial number in IDM, click **Configuration > Licensing**, or in the CLI use the **show version** command.

- Valid Cisco.com username and password

Note You can install the first few signature updates for 5.*x* without a license. This gives you time to get your sensor licensed. If you are unable to get your sensor licensed because of confusion with your contract, you can obtain a 60-day trail license that supports signature updates that require licensing.

You can obtain a license key from the Cisco.com licensing server, which is then delivered to the sensor. Or, you can update the license key from a license key provided in a local file. Go to http://www.cisco.com/go/license and click **IPS Signature Subscription Service** to apply for a license key. For the procedure, see Obtaining and Installing the License Key, page 18-11.

You can view the status of the license key on the Licensing panel in IDM. Whenever you start IDM, you are informed of your license status—whether you have a trial, invalid, or expired license key. With no license key, an invalid license key, or an expired license key, you can continue to use IDM but you cannot download signature updates.

When you enter the CLI, you are informed of your license status. For example, you receive the following message if there is no license installed:

```
***LICENSE NOTICE***
There is no license key installed on the system.
Please go to http://www.cisco.com/go/license to obtain a new license or install a license.
```

You will continue to see this message until you install a license key.

Service Programs for IPS Products

You must have a Cisco Services for IPS service contract for any IPS product so that you can download a license key and obtain the latest IPS signature updates. If you have a direct relationship with Cisco Systems, contact your account manager or service account manager to purchase the Cisco Services for IPS service contract. If you do not have a direct relationship with Cisco Systems, you can purchase the service account from a one-tier or two-tier partner.

When you purchase the following IPS products you must also purchase a Cisco Services for IPS service contract:

- IDS-4215
- IPS-4240
- IPS-4255
- IDSM-2
- NM-CIDS

For ASA products, if you purchased one of the following ASA products that do not contain IPS, you must purchase a SMARTnet contract:

- ASA5510-K8
- ASA5510-DC-K8
- ASA5510-SEC-BUN-K9
- ASA5520-K8
- ASA5520-DC-K8
- ASA5520-BUN-K9
- ASA5540-K8
- ASA5540-DC-K8
- ASA5540-BUN-K9

Note SMARTnet provides operating system updates, access to Cisco.com, access to TAC, and hardware replacement NBD on site.

If you purchased one of the following ASA products that ships with the AIP-SSM installed or if you purchased AIP-SSM to add to your ASA product, you must purchase the Cisco Services for IPS service contract:

- ASA5510-AIP10-K9
- ASA5520-AIP10-K9
- ASA5520-AIP20-K9
- ASA5540-AIP20-K9
- ASA-SSM-AIP-10-K9
- ASA-SSM-AIP-20-K9

Note Cisco Services for IPS provides IPS signature updates, operating system updates, access to Cisco.com, access to TAC, and hardware replacement NBD on site.

For example, if you purchased an ASA-5510 and then later wanted to add IPS and purchased an ASA-SSM-AIP-10-K9, you must now purchase the Cisco Services for IPS service contract.

After you have the Cisco Services for IPS service contract, you must also have your product serial number to apply for the license key. For the procedure, see Obtaining and Installing the License Key, page 18-11.

Caution If you ever send your product for RMA, the serial number will change. You must then get a new license key for the new serial number.

Obtaining and Installing the License Key

This section describes how to obtain and install the license key using IDM or the CLI. It contains the following topics:

Using IDM

Note In addition to a valid Cisco.com username and password, you must also have a Cisco Services for IPS service contract before you can apply for a license key. For more information, see Service Programs for IPS Products, page 18-10.

To obtain and install the sensor license, follow these steps:

Step 1 Log in to IDM using an account with administrator privileges.

Step 2 Choose **Configuration > Licensing**.

The Licensing pane appears. Information about the current license state is displayed. If you have already installed your license, you can click **Download** to update it if needed.

Step 3 Choose the method to deliver the license:

a. Select **Cisco Connection Online** to obtain the license from Cisco.com.

IDM contacts the license server on Cisco.com and sends the server the serial number to obtain the license key. This is the default method. Go to Step 4.

b. Select **License File** to use a license file.

To use this option, you must apply for a license at www.cisco.com/go/license.

The license is sent to you in e-mail and you save it to a drive that is accessible by IDM. This option is useful if your computer does not have access to Cisco.com.

Go to Step 7.

Step 4 Click **Update License**. The Licensing dialog box appears.

Step 5 Click **Yes** to continue. The Status dialog box informs you that the sensor is trying to connect to Cisco.com. An Information dialog box confirms that the license has been updated.

Step 6 Click **OK**.

Step 7 Go to www.cisco.com/go/license.

Step 8 Fill in the required fields.

Caution You must have the correct IPS device serial number because the license key only functions on the device with that number.

Your Cisco IPS Signature Subscription Service license key is sent by e-mail to the e-mail address you specified.

Step 9 Save the license file to a hard-disk drive or a network drive that is accessible by the client running IDM.

Step 10 Log in to IDM.

Step 11 Choose **Configuration > Licensing**.

Step 12 Under Update License, choose **Update From: License File**.

Step 13 In the **Local File Path** field, specify the path to the license file or click **Browse Local** to browse to the file. The Select License File Path dialog box appears.

Step 14 Browse to the license file and click **Open**.

Step 15 Click **Update License**.

Using the CLI

Use the **copy** *source_url license_file_name* **license-key** command to copy the license file to your sensor.

The following options apply:

- *source-url*—The location of the source file to be copied. It can be a URL or keyword.
- *destination-url*—The location of the destination file to be copied. It can be a URL or a keyword.
- **license-key**—The subscription license file.
- *license_file_name*—The name of the license file you receive.

Note You cannot install an older license key over a newer license key.

The exact format of the source and destination URLs varies according to the file. Here are the valid types:

- ftp:—Source or destination URL for an FTP network server. The syntax for this prefix is:

  ftp:[//[username@] location]/relativeDirectory]/filename

  ftp:[//[username@]location]//absoluteDirectory]/filename
- scp:—Source or destination URL for the SCP network server. The syntax for this prefix is:

  scp:[//[username@] location]/relativeDirectory]/filename

  scp:[//[username@] location]//absoluteDirectory]/filename
- http:—Source URL for the web server. The syntax for this prefix is:

  http:[[/[username@]location]/directory]/filename

- https:—Source URL for the web server. The syntax for this prefix is:

  https:[[/[username@]location]/directory]/filename

Note If you use FTP or SCP, you are prompted for a password.

Note If you use SCP, the remote host must be on the SSH known hosts list. For the procedure, see Adding Hosts to the Known Hosts List, page 4-32.

Note If you use HTTPS, the remote host must be a TLS trusted host. For the procedure, see Adding TLS Trusted Hosts, page 4-37.

To install the license key, follow these steps:

Step 1 Apply for the license key at www.cisco.com/go/license.

Step 2 Fill in the required fields.

Note You must have the correct IPS device serial number because the license key only functions on the device with that number.

Your Cisco IPS Signature Subscription Service license key is sent by e-mail to the e-mail address you specified.

Step 3 Save the license key to a system that has a web server, FTP server, or SCP server.

Step 4 Log in to the CLI using an account with administrator privileges.

Step 5 Copy the license key to the sensor:

```
sensor# copy scp://user@10.89.147.3://tftpboot/dev.lic license-key
Password: *******
```

Step 6 Verify the sensor is licensed:

```
sensor# show version
Application Partition:

Cisco Intrusion Prevention System, Version 5.0(1)S149.0

OS Version 2.4.26-IDS-smp-bigphys
Platform: IPS-4255-K9
Serial Number: JAB0815R0JS
Licensed, expires: 19-Dec-2005 UTC
Sensor up-time is 2 days.
Using 706699264 out of 3974291456 bytes of available memory (17% usage)
system is using 17.3M out of 29.0M bytes of available disk space (59% usage)
application-data is using 36.5M out of 166.8M bytes of available disk space (23% usage)
boot is using 39.4M out of 68.6M bytes of available disk space (61% usage)


MainApp          2005_Feb_18_03.00   (Release)   2005-02-18T03:13:47-0600   Running
AnalysisEngine   2005_Feb_15_03.00   (QATest)    2005-02-15T12:59:35-0600   Running
CLI              2005_Feb_18_03.00   (Release)   2005-02-18T03:13:47-0600
```

```
Upgrade History:

    IDS-K9-maj-5.0-1-   14:16:00 UTC Thu Mar 04 2004

Recovery Partition Version 1.1 - 5.0(1)S149


sensor#
```

Step 7 Copy your license key from a sensor to a server to keep a backup copy of the license:

```
sensor# copy license-key scp://user@10.89.147.3://tftpboot/dev.lic
Password: *******
sensor#
```

Cisco Security Intelligence Operations

The Cisco Security Intelligence Operations site on Cisco.com provides intelligence reports about current vulnerabilities and security threats. It also has reports on other security topics that help you protect your network and deploy your security systems to reduce organizational risk.

You should be aware of the most recent security threats so that you can most effectively secure and manage your network. Cisco Security Intelligence Operations contains the top ten intelligence reports listed by date, severity, urgency, and whether there is a new signature available to deal with the threat.

Cisco Security Intelligence Operations contains a Security News section that lists security articles of interest. There are related security tools and links.

You can access Cisco Security Intelligence Operations at this URL:

http://tools.cisco.com/security/center/home.x

Cisco Security Intelligence Operations is also a repository of information for individual signatures, including signature ID, type, structure, and description.

You can search for security alerts and signatures at this URL:

http://tools.cisco.com/security/center/search.x

Accessing IPS Documentation

You can find IPS documentation at this URL:

http://www.cisco.com/en/US/products/hw/vpndevc/ps4077/tsd_products_support_series_home.html

Or to access IPS documentation from Cisco.com, follow these steps:

Step 1 Log in to Cisco.com.

Step 2 Click **Support**.

Step 3 Under Support at the bottom of the page, click **Documentation**.

Step 4 Choose **Products > Security > Intrusion Prevention System (IPS) > IPS Appliances > Cisco IPS 4200 Series Sensors**. The Cisco IPS 4200 Series Sensors page appears. All of the most up-to-date IPS documentation is on this page.

Note Although you will see references to other IPS documentation sites on Cisco.com, this is the site with the most complete and up-to-date IPS documentation.

Step 5 Click one of the following categories to access Cisco IPS documentation:

- **Download Software**—Takes you to the Download Software site.

  Note You must be logged into Cisco.com to access the software download site.

- **Release and General Information**—Contains documentation roadmaps and release notes.
- **Reference Guides**—Contains command references and technical references.
- **Design**—Contains design guide and design tech notes.
- **Install and Upgrade**—Contains hardware installation and regulatory guides.
- **Configure**—Contains configuration guides for IPS CLI, IDM, and IME.
- **Troubleshoot and Alerts**—Contains TAC tech notes and field notices.

APPENDIX A

System Architecture

This appendix describes the system architecture of IPS 5.1. It contains the following sections:

System Overview

You can install Cisco IPS software on two platforms: the appliances and the modules (for a list of current appliances and modules, refer to Supported Sensors).

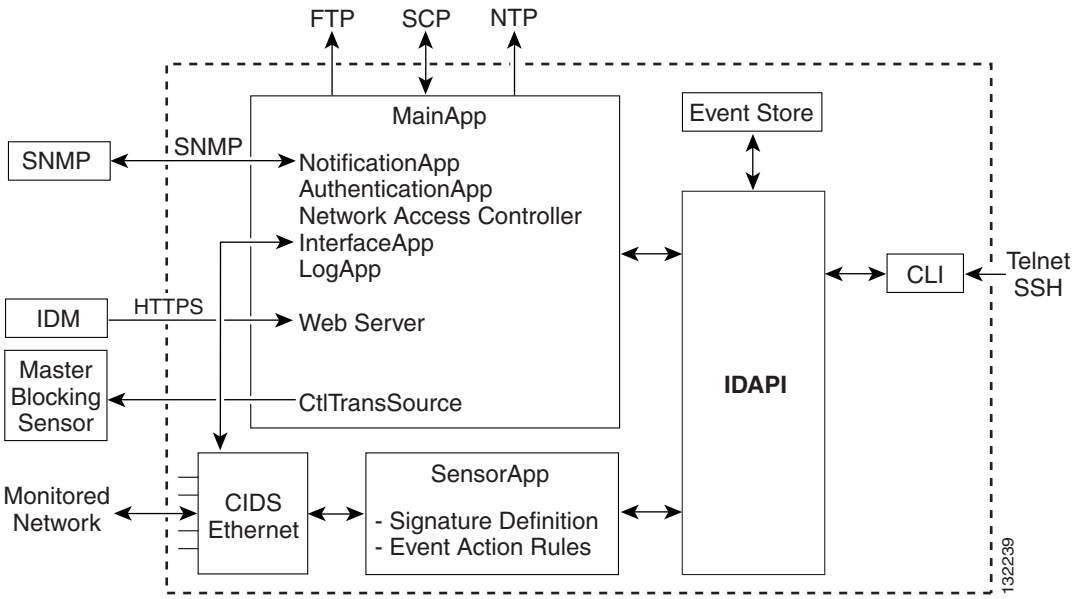
This section contains the following topics:

System Design

IPS software runs on the Linux operating system. We have hardened the Linux OS by removing unnecessary packages from the OS, disabling unused services, restricting network access, and removing access to the shell.

Figure A-1 illustrates the system design:

Figure A-1 System Design



IPS software includes the following applications:

Note Each application has its own configuration file in XML format.

- MainApp—Initializes the system, starts and stops the other applications, configures the OS, and performs upgrades. It contains the following components:
  - ctlTransSource (Control Transaction server)—Allows sensors to send control transactions. This is used to enable Attack Response Controller's (formerly known as Network Access Controller) master blocking sensor capability.
  - Event Store—An indexed store used to store IPS events (error, status, and alert system messages) that is accessible through the CLI, IDM, ASDM, or RDEP.
  - InterfaceApp—Handles bypass and physical settings and defines paired interfaces. Physical settings are speed, duplex, and administrative state.
  - LogApp—Writes all the application's log messages to the log file and the application's error messages to the Event Store.
  - Attack Response Controller (formerly known as Network Access Controller) —Manages remote network devices (firewalls, routers, and switches) to provide blocking capabilities when an alert event has occurred. ARC creates and applies ACLs on the controlled network device or uses the **shun** command (firewalls).
  - NotificationApp—Sends SNMP traps when triggered by alert, status, and error events. NotificationApp uses the public domain SNMP agent. SNMP GETs provide information about the general health of the sensor.

  - Web Server (HTTP RDEP2 server)—Provides a web interface and communication with other IPS devices through RDEP2 using several servlets to provide IPS services.
  - AuthenticationApp—Verifies that users are authorized to perform CLI, IDM, ASDM, or RDEP actions.
- SensorApp (Analysis Engine)—Performs packet capture and analysis.
- CLI—The interface that is run when you successfully log in to the sensor through Telnet or SSH. All accounts created through the CLI will use the CLI as their shell (except the service account—only one service account is allowed). Allowed CLI commands depend on the privilege of the user.

All IPS applications communicate with each other through a common API called IDAPI. Remote applications (other sensors, management applications, and third-party software) communicate with sensors through RDEP2 and SDEE protocols.

The sensor has the following partitions:

- Application partition—A full IPS system image.
- Maintenance partition—A special purpose IPS image used to reimage the application partition of the IDSM-2. When you reimage the maintenance partition, all configuration settings are lost.
- Recovery partition—A special purpose image used for recovery of the sensor. Booting into the recovery partition enables you to completely reimage the application partition. Network settings are preserved, but all other configuration is lost.

IPS 5.1 New Features

Cisco IPS 5.1 contains the following new features:

- Support for the Incident Control System (ICS).

  The ICS service augments Cisco's current IPS Signature Service by providing for the delivery of a more rapid and focused response to breaking threats.
- Inline VLAN (on a stick)

  The sensor can perform inline sensing between one or more VLAN pairs on a single sensor interface. Cisco Catalyst line cards that connect directly to the switch backplane and appliances that connect to an external port of the switch can use this feature.
- Rate Limiting

  A rate limit restricts the amount of a specified type of traffic that is allowed on a network device interface to a percentage of maximum bandwidth capacity. Traffic that exceeds this percentage is dropped by the network device. A rate limit can restrict traffic to a specified target host, or to all traffic that crosses the configured interface/directions. You can rate limit permanently or for a specified amount of time. A rate limit is identified by a protocol, an optional destination address, and an optional data value.
- New Event Actions

  Two new deny attacker event actions have been added in the 5.1 release: Deny Attacker Service Pair Inline and Deny Attacker Victim Pair Inline. A new Request Rate Limit event action with a parameter that lets you specify a percentage of traffic from a denied attacker has been added to support rate limiting.
- GRE/IPV4-in-IPV4 Tunneling

  IPS 5.1 sensors can now monitor GRE and IPV4-inIPV4 encapsulated traffic.

User Interaction

You interact with IPS 5.1 in the following ways:

- Configure device parameters

  You generate the initial configuration for the system and its features. This is an infrequent task, usually done only once. The system has reasonable default values to minimize the number of modifications you must make. You can configure IPS 5.1 through the CLI, IDM, IDS MC, ASDM or through another application using RDEP2/IDCONF.

- Tune

  You make minor modifications to the configuration, primarily to the Analysis Engine, which is the portion of the application that monitors network traffic. You can tune the system frequently after initially installing it on the network until it is operating efficiently and only producing information you find useful. You can create custom signatures, enable features, or apply a service pack or signature update. You can tune IPS 5.1 through the CLI, IDM, IDS MC, ASDM or through another application using RDEP2/IDCONF.

- Update

  You can schedule automatic updates or apply updates immediately to the applications and signature data files. You can update IPS 5.1 through the CLI, IDM, IDS MC, ASDM or through another application using RDEP2/IDCONF

- Retrieve information

  You can retrieve data (status messages, errors, and alerts) from the system through the CLI, IDM, IDS MC, ASDM or another application using RDEP or RDEP2.

Security Features

IPS 5.1 has the following security features:

- Network access is restricted to hosts who are specifically allowed access.
- All remote hosts who attempt to connect through Web Server, SSH and SCP or Telnet will be authenticated.
- By default Telnet access is disabled. You can choose to enable Telnet.
- By default SSH access is enabled.
- An FTP server does not run on the sensor. You can use SCP to remotely copy files.
- By default Web Server uses TLS or SSL. You can choose to disable TLS and SSL.
- Unnecessary services are disabled.
- Only the SNMP set required by the Cisco MIB Police is allowed within the CISCO-CIDS-MIB. OIDs implemented by the public domain SNMP agent will be writeable when specified by the MIB.

MainApp

MainApp now includes all IPS components except SensorApp and the CLI. This section describes MainApp, and contains the following topics:

MainApp Responsibilities

MainApp has the following responsibilities:

- Validate the Cisco-supported hardware platform
- Report software version and PEP information
- Start, stop, and report the version of the IPS components
- Configure the host system settings
- Manage the system clock
- Manage the Event Store
- Install and uninstall software upgrades
- Shut down or reboot the operating system

MainApp responds to the **show version** command by displaying the following information:

- Sensor build version
- MainApp version
- Version of each running application
- Version and timestamp of each installed upgrade
- Next downgrade version of each installed upgrade
- Platform version (for example, IDS-4240, WS-SVC-IDSM2)
- Version of sensor build on the other partition

MainApp also gathers the host statistics.

The following applications are now part of MainApp and are responsible for event storage, management, actions, and communication: Event Store, NotificationApp, CtlTransSource, ARC (formerly known as Network Access Controller), and LogApp.

These applications contain the following new features:

- SNMP support through NotificationApp

  Support for SNMP is one of the most significant changes for the management interface of the system. Through SNMP you can obtain standard health and welfare information about the system. Signatures have a new action of SNMP notification that causes an SNMP trap to be sent when the signatures fires.

  SNMP version 2 is the only version of SNMP supported.

- Event storage and retrieval

  The oldest entries expire in Event Store when there is no more room for new entries. RDEP provides different queries for retrieving just audit data vs. IPS alert data. All RDEP and RDEP2 SDEE queries are supported. All events are stored in SDEE CIDEE format.

- New "health" control transaction

  A new health and welfare type of control transaction is defined in the IDCONF specification. This control transaction reports the status and welfare of the system.

Event Store

This section describes Event Store, and contains the following topics:

- About Event Store, page A-6
- Event Data Structures, page A-7
- IPS Events, page A-8

About Event Store

Each IPS event is stored in Event Store with a time stamp and a unique, monotonic, ascending ID. This time stamp is the primary key used to index the event into the fixed-size, indexed Event Store. When the circular Event Store has reached its configured size, the oldest event or events are overwritten by the new event being stored. SensorApp is the only application that writes alert events into the Event Store. All applications write log, status, and error events into the Event Store.

The fixed-sized, indexed Event Store allows simple event queries based on the time, type, priority, and a limited number of user-defined attributes. If each event is assigned a priority of low, medium, or high, a single event query can specify a list of desired event types, intrusion event priorities, and a time range.

Table A-1 shows some examples:

Table A-1 IPS Event Examples

IPS Event Type	Intrusion Event Priority	Start Time Stamp Value	Stop Time Stamp Value	Meaning
status	—	0	Maximum value	Get all status events that are stored.
error status	—	0	65743	Get all error and status events that were stored before time 65743.
status	—	65743	Maximum value	Get status events that were stored at or after time 65743.

Table A-1 IPS Event Examples (continued)

IPS Event Type	Intrusion Event Priority	Start Time Stamp Value	Stop Time Stamp Value	Meaning
intrusion attack response	low	0	Maximum value	Get all intrusion and attack response events with low priority that are stored.
attack response error status intrusion	medium high	4123000000	4123987256	Get attack response, error, status, and intrusion events with medium or high priority that were stored between time 4123000000 and 4123987256.

The size of the Event Store allows sufficient buffering of the IPS events when the sensor is not connected to an IPS event consumer. Sufficient buffering depends on your requirements and the capabilities of the nodes in use. The oldest events in the circular buffer are replaced by the newest events.

Event Data Structures

The various functional units communicate the following seven types of data:

- Intrusion events—Produced by SensorApp. The sensor detects intrusion events.
- Error events—Caused by hardware or software malfunctions.
- Status events—Reports of a change in the application's status, for example, that its configuration has been updated.
- Control transaction log events—The sensor logs the result of a control transaction.
- Attack response events—Actions for the ARC, for example, a block request.
- Debug events—Highly detailed reports of a change in the application's status used for debugging.
- Control transaction data—Data associated with control transactions, for example, diagnostic data from an application, session logs, and configuration data to or from an application.

All seven types of data are referred to collectively as *IPS data*. The six event types—intrusion, error, status, control transaction log, network access, and debug—have similar characteristics and are referred to collectively as *IPS events*. IPS events are produced by the several different applications that make up the IPS and are subscribed to by other IPS applications. IPS events have the following characteristics:

- They are spontaneously generated by the application instances configured to do so. There is no request from another application instance to generate a particular event.
- They have no specific destination. They are stored and then retrieved by one or more application instances.

Control transactions involve the following types of requests:

- Request to update an application instance's configuration data
- Request for an application instance's diagnostic data
- Request to reset an application instance's diagnostic data
- Request to restart an application instance
- Request for ARC, such as a block request

Control transactions have the following characteristics:

- They always consist of a request followed by a response.

  The request and response may have an arbitrary amount of data associated with them. The response always includes at least a positive or negative acknowledgment.

- They are point-to-point transactions.

  Control transactions are sent by one application instance (the initiator) to another application instance (the responder).

IPS data is represented in XML format as an XML document. The system stores user-configurable parameters in several XML files.

IPS Events

IPS applications generate IPS events to report the occurrence of some stimulus. The events are the data, such as the alerts generated by SensorApp or errors generated by any application. Events are stored in a local database known as the Event Store.

There are five types of events:

- evAlert—Alert event messages that report when a signature is triggered by network activity.
- evStatus—Status event messages that report the status and actions of the IPS applications.
- evError— Error event messages that report errors that occurred while attempting response actions.
- evLogTransaction—Log transaction messages that report the control transactions processed by each sensor application.
- evShunRqst—Block request messages that report when ARC issues a block request.

You can view the status and error messages using the CLI, IDM, and ASDM.

SensorApp and ARC log response actions (TCP resets, IP logging start and stop, blocking start and stop, trigger packet) as status messages.

NotificationApp

NotificationApp allows the sensor to send alerts and system error messages as SNMP traps. It subscribes to events in the Event Store and translates them into SNMP MIBs and sends them to destinations through a public-domain SNMP agent. NotificationApp supports sending sets and gets. The SNMP GETs provide information about basic sensor health.

NotificationApp sends the following information from the evAlert event in sparse mode:

- Originator information
- Event ID
- Event severity
- Time (UTC and local time)
- Signature name
- Signature ID
- Subsignature ID
- Participant information
- Alarm traits

NotificationApp sends the following information from the evAlert event in detail mode:

- Originator information
- Event ID
- Event severity
- Time (UTC and local time)
- Signature name
- Signature ID
- Subsignature ID
- Version
- Summary
- Interface group
- VLAN
- Participant information
- Actions
- Alarm traits
- Signature
- IP log IDs

NotificationApp determines which evError events to send as a trap according to the filter that you define. You can filter based on error severity (error, fatal, and warning). NotificationApp sends the following information from the evError event:

- Originator information
- Event ID
- Event severity
- Time (UTC and local time)
- Error message

NotificationApp supports GETs for the following general health and system information from the sensor:

- Packet loss
- Packet denies
- Alarms generated
- Fragments in FRP
- Datagrams in FRP
- TCP streams in embryonic state
- TCP streams in established state
- TCP streams in closing state
- TCP streams in system
- TCP packets queued for reassembly
- Total nodes active

- TCP nodes keyed on both IP addresses and both ports
- UDP nodes keyed on both IP addresses and both port
- IP nodes keyed on both IP address
- Sensor memory critical stage
- Interface status
- Command and control packet statistics
- Fail-over state
- System uptime
- CPU usage
- Memory usage for the system
- PEP

Note Not all IDS and IPS platforms support PEP.

NotificationApp provides the following statistics:

- Number of error traps
- Number of event action traps
- Number of SNMP GET requests
- Number of SNMP SET requests

CtlTransSource

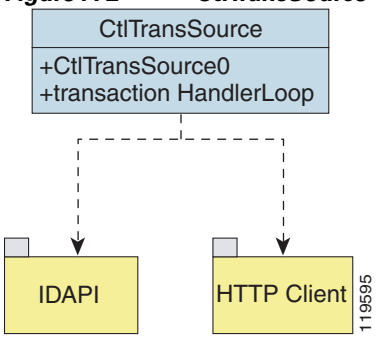
CtlTransSource is an application that forwards locally initiated remote control transactions to their remote destinations using the RDEP and HTTP protocols. CtlTransSource initiates either TLS or non-TLS connections and communicates remote control transactions to HTTP servers over these connections.

CtlTransSource must establish sufficient credentials on the remote HTTP server to execute a remote control transaction. It establishes its credentials by presenting an identity to the HTTP server on the remote node in the form of a username and password (basic authentication). When the authentication is successful, the requestor is assigned a cookie containing a user authentication that must be presented with each request on that connection.

The transactionHandlerLoop method in the CtlTransSource serves as a proxy for remote control transaction. When a local application initiates a remote control transaction, IDAPI initially directs the transaction to CtlTransSource. The transactionHandlerLoop method is a loop that waits on remote control transactions that are directed to CtlTransSource.

Figure A-2 shows the transactionHandlerLoop method in the CtlTransSource.

Figure A-2 CtlTransSource



When the transactionHandlerLoop receives a remotely addressed transaction, it tries to forward the remote control transaction to its remote destination. The transactionHandlerLoop formats the transaction into an RDEP control transaction message. The transactionHandlerLoop uses the HttpClient classes to issue the RDEP control transaction request to the HTTP server on the remote node. The remote HTTP server handles the remote control transaction and returns the appropriate RDEP response message in an HTTP response. If the remote HTTP server is an IPS web server, the web server uses the CtlTransSource servlet to process the remote control transactions.

The transactionHandlerLoop returns either the RDEP response or a failure response as the control transaction's response to the remote control transaction's initiator. If the HTTP server returns an unauthorized status response (indicating the HTTP client has insufficient credentials on the HTTP server), the transactionHandlerLoop reissues the transaction request using CtlTransSource's designated username and password to authenticate the requestor's identity. The transactionHandlerLoop continues to loop until it receives a control transaction that directs it to exit or until its exit event is signaled.

Attack Response Controller

This section describes ARC, which is the IPS application that starts and stops blocking on routers, switches, and firewalls. A *block* is an entry in a device's configuration or ACL to block incoming and outgoing traffic for a specific host IP address or network address. ARC also controls rate limiting on routers running Cisco IOS 12.3.

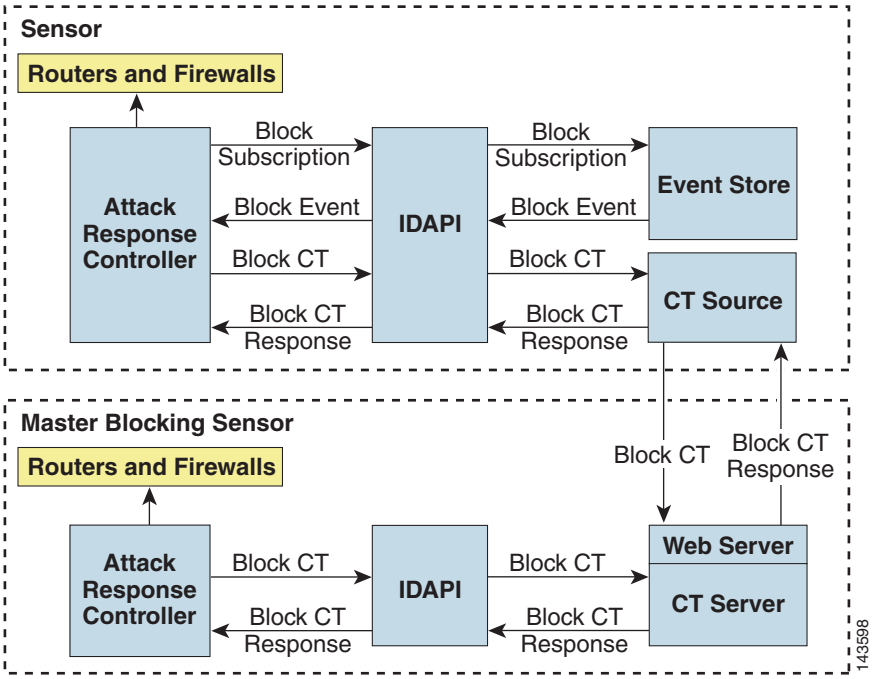
This section contains the following topics:

About ARC

ARC's main responsibility is to block events. When it responds to a block, it either interacts with the devices it is managing directly to enable the block or it sends a block request through the Control Transaction Server to a master blocking sensor. The Web Server on the master blocking sensor receives the control transaction and passes it to the Control Transaction Server, which passes it to ARC. ARC on the master blocking sensor then interacts with the devices it is managing to enable the block.

Figure A-3 illustrates ARC.

Figure A-3 ARC



Note An ARC instance can control 0, 1, or many network devices. ARC does not share control of any network device with other ARC applications, IPS management software, other network management software, or system administrators. Only one ARC instance is allowed to run on a given sensor.

ARC initiates a block in response to one of the following:

- An alert event generated from a signature that is configured with a block action
- A block configured manually through the CLI, IDM, or ASDM
- A block configured permanently against a host or network address

When you configure ARC to block a device, it initiates either a Telnet or SSH connection with the device. ARC maintains the connection with each device. After the block is initiated, ARC pushes a new set of configurations or ACLs (one for each interface direction) to each controlled device. When a block is completed, all configurations or ACLs are updated to remove the block.

ARC Features

ARC has the following features:

- Communication through Telnet and SSH 1.5 with 3DES (the default) or DES encryption

  Only the protocol specified in the ARC configuration for that device is attempted. If the connection fails for any reason, ARC attempts to reestablish it.

- Preexisting ACLs on routers and VACLs on switches

  If a preexisting ACL exists on a router interface or direction that is controlled by ARC, you can specify that this ACL be merged into the ARC-generated configuration, either before any blocks by specifying a preblock ACL or after any blocks by specifying a postblock ACL. The Catalyst 6000 VACL device types can have a preblock and postblock VACL specified for each interface that ARC controls. The firewall device types use a different API to perform blocks and ARC does not have any effect on preexisting ACLs on the firewalls.

  Note Catalyst 5000 RSM and Catalyst 6000 MSFC2 network devices are supported in the same way as Cisco routers.

  For more information, see ACLs and VACLs, page A-15.

- Forwarding blocks to a list of remote sensors

  ARC can forward blocks to a list of remote sensors, so that multiple sensors can in effect collectively control a single network device. Such remote sensors are referred to as master blocking sensors. For more information on master blocking sensors, see Configuring the Sensor to be a Master Blocking Sensor, page 10-28.

- Specifying blocking interfaces on a network device

  You can specify the interface and direction where blocking is performed in the ARC configuration for routers. You can specify the interface where blocking is performed in the VACL configuration.

  Note Cisco firewalls do not block based on interface or direction, so this configuration is never specified for them.

  ARC can simultaneously control up to 250 interfaces.

- Blocking hosts or networks for a specified time

  ARC can block a host or network for a specified number of minutes or indefinitely. ARC determines when a block has expired and unblocks the host or network at that time.

- Logging important events

  ARC writes a confirmation event when block or unblock actions are completed successfully or if any errors occur. ARC also logs important events such as loss and recovery of a network device communication session, configuration errors, and errors reported by the network device.

- Maintaining the blocking state across ARC restarts

  ARC reapplies blocks that have not expired when a shutdown or restart occurs. ARC removes blocks that have expired while it was shut down.

  Note ARC can only maintain the blocking state successfully if no one changes the system time while the application is shut down.

  For more information, see Maintaining State Across Restarts, page A-15.

- Maintaining blocking state across network device restarts

ARC reapplies blocks and removes expired blocks as needed whenever a network device is shut down and restarted. ARC is not affected by simultaneous or overlapping shutdowns and restarts of ARC.

- Authentication and authorization

  ARC can establish a communications session with a network device that uses AAA authentication and authorization including the use of remote TACACS+ servers.

- Two types of blocking

  ARC supports host blocks and network blocks. Host blocks are connection based or unconditional. Network blocks are always unconditional.

  For more information, see Connection-Based and Unconditional Blocking, page A-16.

- NAT addressing

  ARC can control network devices that use a NAT address for the sensor. If you specify a NAT address when you configure a network device, that address is used instead of the local IP address when the sensor address is filtered from blocks on that device.

- Single point of control

  ARC does not share control of network devices with administrators or other software. If you must update a configuration, shut down ARC until the change is complete. You can enable or disable ARC through the CLI or any IPS manager. When ARC is reenabled, it completely reinitializes itself, including rereading the current configuration for each controlled network device.

  Note We recommend that you disable ARC from blocking when you are configuring any network device, including firewalls.

- Maintains up to 250 active blocks at any given time

  ARC can maintain up to 250 active blocks at a time. Although ARC can support up to 65535 blocks, we recommend that you allow no more than 250 at a time.

  Note The number of blocks is not the same as the number of interface and directions.

Supported Blocking Devices

ARC can control the following devices:

- Cisco routers running Cisco IOS 11.2 or later

  Note To perform rate limiting, the routers must be running Cisco IOS 12.3 or later.

- Catalyst 5000 series switches with Supervisor Engine software 5.3(1) or later running on the supervisor engine, and IOS 11.2(9)P or later running on the RSM.

  Note You must have the RSM because blocking is performed on the RSM.

- Catalyst 6000 series switches with PFC installed running Catalyst software 5.3 or later

- Catalyst 6000 MSFC2 with Catalyst software 5.4(3) or later and Cisco IOS 12.1(2)E or later on the MSFC2
- Cisco ASA 500 series models: ASA 5510, ASA 5520, and ASA 5540
- FWSM

Note The FWSM cannot block in multi-mode admin context.

ACLs and VACLs

If you want to filter packets on an interface or direction that ARC controls, you can configure ARC to apply an ACL before any blocks (preblock ACL) and to apply an ACL after any blocks (postblock ACL). These ACLs are configured on the network device as inactive ACLs. You can define preblock and postblock ACLs for each interface and direction. ARC retrieves and caches the lists and merges them with the blocking ACEs whenever it updates the active ACL on the network device. In most cases, you will want to specify a preexisting ACL as the postblock ACL so that it does not prevent any blocks from taking effect. ACLs work by matching a packet to the first ACE found. If this first ACE permits the packet, a subsequent deny statement will not be found.

You can specify different preblock and postblock ACLs for each interface and direction, or you can reuse the same ACLs for multiple interfaces and directions. If you do not want to maintain a preblock list, you can use the never block option and always block hosts and networks by using existing configuration statements. A forever block is a normal block with a timeout value of -1.

ARC only modifies ACLs that it owns. It does not modify ACLs that you have defined. The ACLs maintained by ARC have a specific format that should not be used for user-defined ACLs. The naming convention is **IPS_<**interface_name**>_[in | out]_[0 | 1]**. <interface_name> corresponds to the name of the blocking interface as given in the ARC configuration.

For Catalyst switches, it is a blocking interface VLAN number. Do not use these names for preblock and postblock ACLs.

For Catalyst 6000 VACLs, you can specify a preblock and postblock VACL and only the interface is specified (direction is not used in VLANs).

For firewalls, you cannot use preblock or postblock ACLs because the firewall uses a different API for blocking. Instead you must create ACLs directly on the firewalls. For more information, see Blocking with Cisco Firewalls, page A-17.

Maintaining State Across Restarts

When the sensor shuts down, ARC writes all blocks and rate limits (with starting timestamps) to a local file (nac.shun.txt) that is maintained by ARC. When ARC starts, this file is used to determine if any block updates should occur at the controlled network devices. Any unexpired blocks found in the file are applied to the network devices at startup. When ARC shuts down, no special actions on the ACLs are taken even if outstanding blocks are in effect. The nac.shun.txt file is accurate only if the system time is not changed while ARC is not running.

Caution Do not make manual changes to the nac.shun.txt file.

The following scenarios demonstrate how ARC maintains state across restarts.

Scenario 1

There are two blocks in effect when ARC stops and one of them expires before ARC restarts. When ARC restarts, it first reads the nac.shun.txt file. It then reads the preblock and postblock ACLs or VACLs. The active ACL or VACL is built in the following order:

1. The **allow** *sensor_ ip_address* command (unless the **allow sensor shun** command has been configured)
2. Preblock ACL
3. The **always block** command entries from the configuration
4. Unexpired blocks from nac.shun.txt
5. Postblock ACL

When a host is specified as never block in the ARC configuration, it does not get translated into permit statements in the ACL. Instead, it is cached by ARC and used to filter incoming addShunEvent events and addShunEntry control transactions.

Scenario 2

There are no preblock or postblock ACLs specified, but there is an existing active ACL. The new ACL is built in the following order:

1. The **allow** *sensor_ ip_address* command (unless the **allow sensor shun** command has been configured)
2. The **always block** command entries from the configuration
3. Unexpired blocks from nac.shun.txt
4. The **permit IP any any** command

Connection-Based and Unconditional Blocking

ARC supports two types of blocking for hosts and one type of blocking for networks. Host blocks are connection-based or unconditional. Network blocks are always unconditional.

When a host block is received, ARC checks for the connectionShun attribute on the host block. If connectionShun is set to true, ARC performs connection blocking. Any host block can contain optional parameters, such as destination IP address, source port, destination port, and protocol. For a connection block to take place, at least the source and destination IP address must be present. If the source port is present on a connection block, it is ignored and not included in the block.

Under the following conditions, ARC forces the block to be unconditional, converting the block from connection type if necessary:

- A block of any type is active for a specified source IP address
- A new block of any type is received for that source IP address
- The new block differs in any of its optional parameters (except the source port) from the old block

When a block is updated (for example, when a new block arrives while an existing block for that source IP address or network is already in effect), the remaining minutes of the existing block are determined. If the time for the new block is less than or equal to the remaining minutes, no action is taken. Otherwise, the new block timeout replaces the existing block timeout.

Caution Cisco firewalls do not support connection blocking of hosts. When a connection block is applied, the firewall treats it like an unconditional block. Cisco firewalls also do not support network blocking. ARC never tries to apply a network block to a Cisco firewall.

Blocking with Cisco Firewalls

ARC performs blocks on firewalls using the **shun** command. The **shun** command has the following formats:

- To block an IP address:

  shun srcip [*destination_ip_address source_port destination_port* [*port*]]

- To unblock an IP address:

  no shun ip

- To clear all blocks:

  clear shun

- To show active blocks or to show the global address that was actually blocked:

  show shun [*ip_address*]

ARC uses the response to the **show shun** command to determine whether the block was performed.

The **shun** command does not replace existing ACLs, conduits, or outbound commands, so there is no need to cache the existing firewall configuration, nor to merge blocks into the firewall configuration.

Caution Do not perform manual blocks or modify the existing firewall configuration while ARC is running.

If the **block** command specifies only the source IP address, existing active TCP connections are not broken, but all incoming packets from the blocked host are dropped.

When ARC first starts up, the active blocks in the firewall are compared to an internal blocking list. Any blocks that do not have a corresponding internal list entry are removed.

ARC supports authentication on a firewall using local usernames or a TACACS+ server. If you configure the firewall to authenticate using AAA but without the TACACS+ server, ARC uses the reserved username *pix* for communications with the firewall.

If the firewall uses a TACACS+ server for authentication, you use a TACACS+ username. In some firewall configurations that use AAA logins, you are presented with three password prompts: the initial firewall password, the AAA password, and the enable password. ARC requires that the initial firewall password and the AAA password be the same.

When you configure a firewall to use NAT or PAT and the sensor is checking packets on the firewall outside network, if you detect a host attack that originates on the firewall inside network, the sensor tries to block the translated address provided by the firewall. If you are using dynamic NAT addressing, the block can be ineffective or cause innocent hosts to be blocked. If you are using PAT addressing, the firewall could block the entire inside network. To avoid these situations, position your sensor on the inside interface or do not configure the sensor to block.

Blocking with Catalyst Switches

Catalyst switches with a PFC filter packets using VACLs. VACLs filter all packets between VLANs and within a VLAN.

MSFC router ACLs are supported when WAN cards are installed and you want the sensor to control the interfaces through the MSFC2.

Note An MSFC2 card is not a required part of a Catalyst switch configuration for blocking with VACLs.

Caution When you configure ARC for the Catalyst switch, do not specify a direction with the controlled interface. The interface name is a VLAN number. Preblock and postblock lists should be VACLs.

The following commands apply to the Catalyst VACLs:

- To view an existing VACL:

  `show security acl info` *acl_name*

- To block an address (*address_spec* is the same as used by router ACLs):

  `set security acl ip` *acl_name* **`deny`** *address_spec*

- To activate VACLs after building the lists:

  `commit security acl all`

- To clear a single VACL:

  `clear security acl map` *acl_name*

- To clear all VACLs:

  `clear security acl map all`

- To map a VACL to a VLAN:

  `set sec acl` *acl_name vlans*

LogApp

The sensor logs all events (alert, error, status, and debug messages) in a persistent, circular buffer. The sensor also generates IP logs. The messages and IP logs are accessible through the CLI, IDM, ASDM, and RDEP clients.

The IPS applications use LogApp to log messages. LogApp sends log messages at any of five levels of severity: debug, timing, warning, error, and fatal. LogApp writes the log messages to /usr/cids/idsRoot/log/main.log, which is a circular text file. New messages overwrite older messages when the file reaches its maximum size, therefore the last message written may not appear at the end of the main.log. Search for the string "= END OF FILE =" to locate the last line written to the main.log.

The main.log is included in the **show tech-support** command output. If the message is logged at warning level or above (error or fatal), LogApp converts the message to an evError event (with the corresponding error severity) and inserts it in Event Store.

Note For the procedure for displaying tech support information, see Displaying Tech Support Information, page 13-18. For the procedure for displaying events, see Displaying Events, page 13-4.

LogApp receives all syslog messages, except cron messages, that are at the level of informational and above (*.info;cron.none), and inserts them into Event Store as <evErrors> with the error severity set to Warning. LogApp and application logging are controlled through the service logger commands.

LogApp can control what log messages are generated by each application by controlling the logging severity for different logging zones. You would only access the individual-zone-control of the logger service at the request and supervision of a TAC engineer or developer. For troubleshooting purposes, TAC might request that you turn on debug logging. For more information, see Enabling Debug Logging, page C-27.

AuthenticationApp

This section describes AuthenticationApp, and contains the following topics:

AuthenticationApp Responsibilities

AuthenticationApp has the following responsibilities:

- To authenticate a user's identity
- To administer the user's accounts, privileges, keys, and certificates
- To configure which authentication methods are used by AuthenticationApp and other access services on the sensor

Authenticating Users

You must configure authentication on the sensor to establish appropriate security for user access. When you install a sensor, an initial cisco account with an expired password is created. A user with administrative access to the sensor accesses the sensor through the CLI or an IPS manager, such as IDM or ASDM, by logging in to the sensor using the default administrative account (cisco). In the CLI, the Administrator is prompted to change the password. IPS managers initiate a setEnableAuthenticationTokenStatus control transaction to change the account's password.

Through the CLI or an IPS manager, the Administrator configures which authentication method is used, such as username and password or an SSH authorized key. The application servicing the Administrator initiates a setAuthenticationConfig control transaction to establish the authentication configuration.

The authentication configuration includes a login attempt limit value that is used to specify how account locking is handled. Account locking is invoked when the number of consecutive failed login attempts for a given account exceeds the login attempt limit value. After an account is locked, all further attempts to log in to that account are rejected. The account is unlocked by resetting the account's authentication token using the setEnableAuthenticationTokenStatus control transaction. The account locking feature is disabled when the login attempt limit value is set to zero.

The Administrator can add additional user accounts either through the CLI or an IPS manager. For more information, see User Roles, page A-27.

Configuring Authentication on the Sensor

When a user tries to access the sensor through a service such as Web Server or the CLI, the user's identity must be authenticated and the user's privileges must be established. The service that is providing access to the user initiates an execAuthenticateUser control transaction request to AuthenticationApp to authenticate the user's identity. The control transaction request typically includes the username and a password, or the user's identity can be authenticated using an SSH authorized key.

AuthenticationApp responds to the execAuthenticateUser control transaction request by attempting to authenticate the user's identity. AuthenticationApp returns a control transaction response that contains the user's authentication status and privileges. If the user's identity cannot be authenticated, AuthenticationApp returns an unauthenticated status and anonymous user privileges in the control transaction response. The control transaction response also indicates if the account's password has expired. User interface applications that authenticate users by initiating an execAuthenticateUser control transaction prompt the user to change the password.

AuthenticationApp uses the underlying operating system to confirm a user's identity. All the IPS applications send control transactions to AuthenticationApp, which then uses the operating system to form its responses.

Remote shell services, Telnet and SSH, are not IPS applications. They call the operating system directly. If the user is authenticated, it launches the IPS CLI. In this case, the CLI sends a special form of the execAuthenticateUser control transaction to determine the privilege level of the logged-in user. The CLI then tailors the commands it makes available based on this privilege level.

Managing TLS and SSH Trust Relationships

Encrypted communications over IP networks provide data privacy by making it impossible for a passive attacker to discover from the packets exchanged alone the secret key needed to decrypt the data in the packets.

However, an equally dangerous attack vector is for an imposter to pretend to be the server end of the connection. All encryption protocols provide a means for clients to defend themselves from these attacks. IPS supports two encryption protocols, SSH and TLS, and AuthenticationApp helps manage trust when the sensor plays either the client or server role in encrypted communications.

The IPS Web Server and SSH server are server endpoints of encrypted communications. They protect their identities with a private key and offer a public key to clients that connect to them. For TLS this public key is included inside an X.509 certificate, which includes other information. Remote systems that connect to the sensor should verify that the public key received during connection establishment is the key they expect.

Clients must maintain a list of trusted public keys to protect themselves from man-in-the-middle attacks. The exact procedure by which this trust is established varies depending on the protocol and client software. In general, the client displays a fingerprint of 16 or 20 bytes. The human operator who is configuring the client to establish trust should use an out-of-band method to learn the server's key fingerprints before attempting to establish trust. If the fingerprints match, the trust relationship is established and henceforth the client can automatically connect with that server and be confident that the remote server is not an imposter.

You can use the **show ssh server-key** and **show tls fingerprint** to display the sensor's key fingerprints. By recording the output of these commands when directly connected to the sensor console, you can reliably use this information to confirm the sensor's identity over the network later when establishing trust relationships.

For example, when you initially connect to a sensor through the Microsoft Internet Explorer web browser, a security warning dialog box indicates that the certificate is not trusted. Using Internet Explorer's user interface, you can inspect the certificate thumbprint, a value that should exactly match the SHA1 fingerprint displayed by the **show tls fingerprint** command. After verifying this, add this certificate to the browser's list of trusted CAs to establish permanent trust.

Each TLS client has different procedures for establishing this trust. The sensor itself includes a TLS client that is used to send control transactions to other sensors and download upgrades and configuration files from other TLS web servers. Use the **tls trusted-host** command to establish trust of the TLS servers with which the sensor communicates.

Similarly, the sensor includes an SSH client that is used to communicate with managed network devices, download upgrades, and copy configurations and support files to remote hosts. Use the **ssh host-key** command to establish trust relationships with the SSH servers the sensor will contact.

You can manage the list of TLS trusted certificates and SSH known hosts through the commands **service trusted-certificates** and **service ssh-known-hosts**.

X.509 certificates include additional information that can increase the security of the trust relationship; however, these can lead to confusion. For example, an X.509 certificate includes a validity period during which the certificate can be trusted. Typically this period is a number of years starting at the moment the certificate is created. To ensure that an X.509 certificate is valid at the moment it is being used requires that the client system maintain an accurate clock.

X.509 certificates are also tied to a particular network address. Sensors fill this field with the IP address of the sensor's command and control interface. Consequently, if you change the command and control IP address of the sensor, the server's X.509 certificate is regenerated. You must reconfigure all clients on the network that trusted the old certificate to locate the sensor at its new IP address and trust the new certificate.

By using the SSH known hosts and TLS trusted certificates services in AuthenticationApp, you can operate sensors at a high level of security.

Web Server

Web Server provides RDEP2 support, which enables the sensor to report security events, receive IDIOM transactions, and serve IP logs.

Web Server supports HTTP 1.0 and 1.1. Communications with Web Server often include sensitive information, such as passwords, that would severely compromise the security of the system if an attacker were able to eavesdrop. For this reason, sensors ship with TLS enabled. The TLS protocol is an encryption protocol that is compatible with SSL.

SensorApp

This section describes SensorApp, and contains the following topics:

Responsibilities and Components

SensorApp performs packet capture and analysis. Policy violations are detected through signatures in SensorApp and the information about the violations is forwarded to the Event Store in the form of an alert.

Packets flow through a pipeline of processors fed by a producer designed to collect packets from the network interfaces on the sensor.

SensorApp supports the following processors:

- Time Processor (TP)

  This processor processes events stored in a time-slice calendar. Its primary task is to make stale database entries expire and to calculate time-dependent statistics.

- Deny Filters Processor (DFP)

  This processor handles the deny attacker functions. It maintains a list of denied source IP addresses. Each entry in the list expires based on the global deny timer, which you can configure in the virtual sensor configuration.

- Signature Event Action Processor (SEAP)

  This processor processes event actions. It supports the following event actions:

  - Reset TCP flow
  - IP log
  - Deny packets
  - Deny flow
  - Deny attacker
  - Alert
  - Block host
  - Block connection
  - Generate SNMP trap
  - Capture trigger packet

  Event actions can be associated with an event RR threshold that must be surpassed for the actions to take place.

- Statistics Processor (SP)

  This processor keeps track of system statistics such as packet counts and packet arrival rates.

- Layer 2 Processor (L2P)

  This processor processes layer 2-related events. It also identifies malformed packets and removes them from the processing path. You can configure actionable events for detecting malformed packets such as alert, capture packet, and deny packet. The layer 2 processor updates statistics about packets that have been denied because of the policy you have configured.

- Database Processor (DBP)

  This processor maintains the signature state and flow databases.

- Fragment Reassembly Processor (FRP)

  This processor reassembles fragmented IP datagrams. It is also responsible for normalization of IP fragments when the sensor is in inline mode.

- Stream Reassembly Processor (SRP)

  This processor reorders TCP streams to ensure the arrival order of the packets at the various stream-based inspectors. It is also responsible for normalization of the TCP stream. The normalizer engine lets you enable or disable alert and deny actions.

  The TCP SRP normalizer has a hold-down timer, which lets the stream state rebuild after a reconfiguration event. You cannot configure the timer. During the hold-down interval, the system synchronizes stream state on the first packet in a stream that passes through the system. When the hold down has expired, sensorApp enforces your configured policy. If this policy calls for a denial of streams that have not been opened with a 3-way handshake, established streams that were quiescent during the hold-down period will not be forwarded and will be allowed to timeout. Those streams that were synchronized during the hold-down period are allowed to continue.

- Signature Analysis Processor (SAP)

  This processor dispatches packets to the inspectors that are not stream-based and that are configured for interest in the packet in process.

- Slave Dispatch Processor (SDP)

  A process found only on dual CPU systems.

Some of the processors call inspectors to perform signature analysis. All inspectors can call the alarm channel to produce alerts as needed.

SensorApp also supports the following units:

- Analysis Engine

  The analysis engine handles sensor configuration. It maps the interfaces and also the signature and alarm channel policy to the configured interfaces.

- Alarm Channel

  The alarm channel processes all signature events generated by the inspectors. Its primary function is to generate alerts for each event it is passed.

Packet Flow

Packets are received by the NIC and placed in the kernel user-mapped memory space by the IPS-shared driver. The packet is prepended by the IPS header. Each packet also has a field that indicates whether to pass or deny the packet when it reaches SEAP.

The producer pulls packets from the shared-kernel user-mapped packet buffer and calls the process function that implements the processor appropriate to the sensor model. The following orders occur:

- Single processor execution

  TP --> L2P --> DFP --> FRP --> SP --> DBP --> SAP --> SRP --> EAP

- Dual processor execution

  Execution Thread 1 TP --> L2P --> DFP --> FRP --> SP --> DBP --> SAP --> SDP --> | Execution Thread 2 DBP --> SRP --> EAP

SEAP

SEAP coordinates the data flow from the signature event in the alarm channel to processing through the SEAO, the SEAF, and the SEAH. It consists of the following components:

- Alarm channel

  The unit that represents the area to communicate signature events from the Sensor App inspection path to signature event handling.

- Signature event action override (SEAO)

  Adds actions based on the RR value. SEAO applies to all signatures that fall into the range of the configured RR threshold. Each SEAO is independent and has a separate configuration value for each action type. For more information, see Calculating the Risk Rating, page 6-8.

- Signature event action filter (SEAF)

  Subtracts actions based on the signature event's signature ID, addresses, and RR. The input to the SEAF is the signature event with actions possibly added by the SEAO.

  Note The SEAF can only subtract actions, it cannot add new actions.

  The following parameters apply to the SEAF:

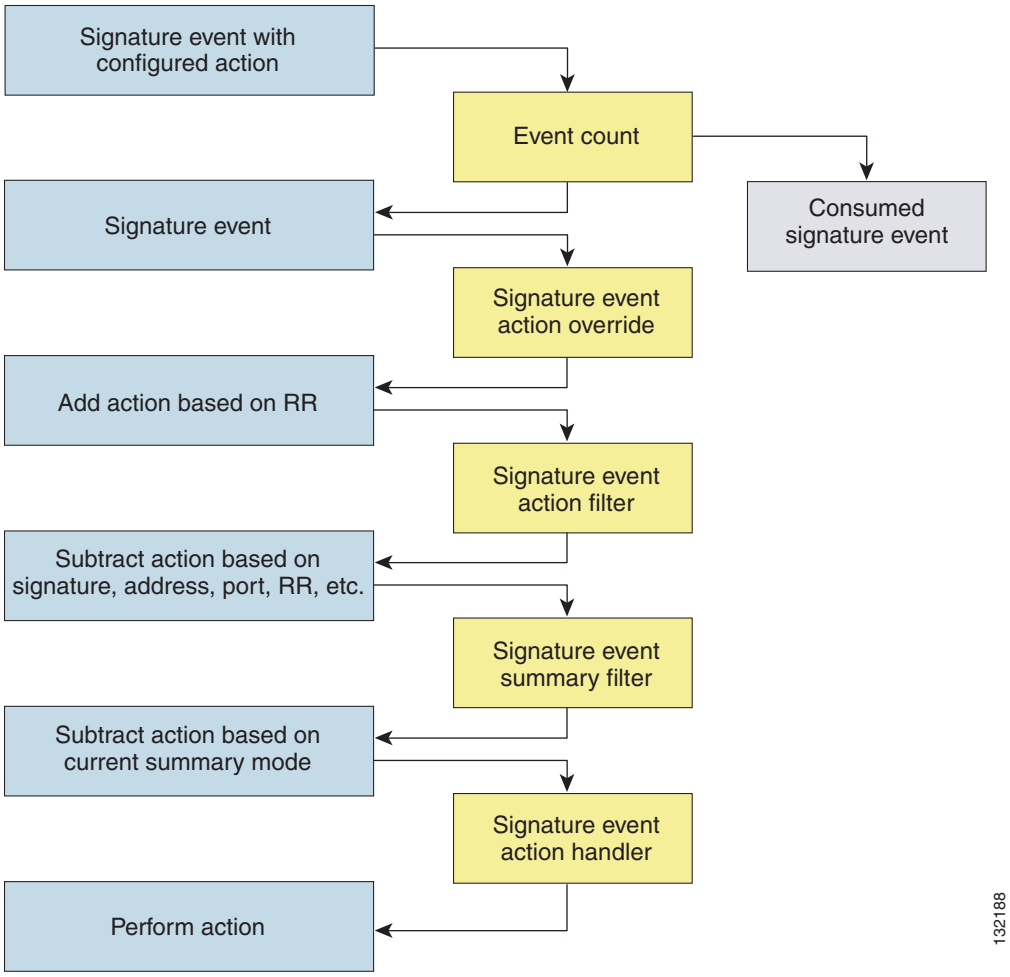
  - Signature ID
  - Subsignature ID
  - Attacker address
  - Attacker port
  - Victim address
  - Victim port
  - RR threshold range
  - Actions to subtract
  - Sequence identifier (optional)
  - Stop-or-continue bit
  - Enable action filter line bit

- Signature event action handler (SEAH)

  Performs the requested actions. The output from the SEAH is the actions being performed and possibly an <evIdsAlert> written to the Event Store.

Figure A-4 on page A-25 illustrates the logical flow of the signature event through the SEAP and the operations performed on the action for this event. It starts with the signature event with configured action received in the alarm channel and flows top to bottom as the signature event passes through the functional components of the SEAP.

Figure A-4 Signature Event Through SEAP



New Features

SensorApp contains the following new features:

- Processing packets inline

  When the sensor is processing packets in the data path, all packets are forwarded without any modifications unless explicitly denied by policy configuration. Because of TCP normalization it is possible that some packets will be delayed to ensure proper coverage. When policy violations are encountered, SensorApp allows for the configuration of actions. Additional actions are available in inline mode, such as deny packet, deny flow, and deny attacker.

  All packets that are unknown or of no interest to the IPS are forwarded to the paired interface with no analysis. All bridging and routing protocols are forwarded with no participation other than a possible deny due to policy violations. There is no IP stack associated with any interface used for inline (or promiscuous) data processing. The current support for 802.1q packets in promiscuous mode is extended to inline mode.

- Enhanced configuration

- Backup for dataflow in inline operations
- Hold down timer

  When SensorApp first starts, it may need to build state information for any flows that currently exist. The hold-down timer prevents SensorApp from denying packets while building this state information. During the hold-down timer, SensorApp still enforces policy whenever there is enough information.
- IP normalization

  Intentional or unintentional fragmentation of IP datagrams can serve to hide exploits making them difficult or impossible to detect. Fragmentation can also be used to circumvent access control policies like those found on firewalls and routers. And different operating systems use different methods to queue and dispatch fragmented datagrams. If the sensor has to check for all possible ways that the end host will reassemble the datagrams, it makes the sensor vulnerable to denial of service attacks. Reassembling all fragmented datagrams inline and only forwarding completed datagrams, refragmenting the datagram if necessary, is the solution to this problem. The IP Fragmentation Normalization unit performs this function.
- TCP normalization

  Through intentional or natural TCP session segmentation, some classes of attacks can be hidden. To make sure policy enforcement can occur with no false positives and false negatives, the state of the two TCP endpoints must be tracked and only the data that is actually processed by the real host endpoints should be passed on. Overlaps in a TCP stream can occur, but are extremely rare except for TCP segment retransmits. Overwrites in the TCP session should not occur. If overwrites do occur, someone is intentionally trying to elude the security policy or the TCP stack implementation is broken. Maintaining full information about the state of both endpoints is not possible unless the sensor acts as a TCP proxy. Instead of the sensor acting as a TCP proxy, the segments will be ordered properly and the normalizer will look for any abnormal packets associated with evasion and attacks.
- Event RR

  The event RR incorporates the following additional information beyond the detection of a potentially malicious action:

  - Severity of the attack if it were to succeed
  - Fidelity of the signature
  - Relevance of the potential attack with respect to the target host
  - Overall value of the target host

  Event RR helps reduce false positives from the system and gives you more control over what causes an alert.
- Event action filters and processing

  4.*x* event filters filtered all actions. 5.*x* event filters handle events separately. Sending the alert is now also considered an action and you can filter or configure it like the other actions.
- Driver support for concurrent SensorApp and TCPdump capture

  The drivers for the data interfaces support concurrent use of the interfaces by SensorApp and TCPdump or other libpcap-based reader

CLI

The CLI provides the sensor user interface for all direct node access such as Telnet, SSH, and serial interface. You configure the sensor applications with the CLI. Direct access to the underlying OS is allowed through the service role.

This section contains the following topics:

User Roles

The CLI for IPS 5.1 permits multiple users to log in at a time. You can create and remove users from the local sensor. You can only modify one user account at a time. Each user is associated with a role that controls what that user can and cannot modify

The CLI supports four user roles: Administrator, Operator, Viewer, and Service. The privilege levels for each role are different; therefore, the menus and available commands vary for each role.

- **Administrators**—This user role has the highest level of privileges. Administrators have unrestricted view access and can perform the following functions:
  - Add users and assign passwords
  - Enable and disable control of physical interfaces and virtual sensors
  - Assign physical sensing interfaces to a virtual sensor
  - Modify the list of hosts allowed to connect to the sensor as a configuring or viewing agent
  - Modify sensor address configuration
  - Tune signatures
  - Assign configuration to a virtual sensor
  - Manage routers
- **Operators**—This user role has the second highest level of privileges. Operators have unrestricted view access and can perform the following functions:
  - Modify their passwords
  - Tune signatures
  - Manage routers
  - Assign configuration to a virtual sensor
- **Viewers**—This user role has the lowest level of privileges. Viewers can view configuration and event data and can modify their passwords.

Tip Monitoring applications only require viewer access to the sensor. You can use the CLI to set up a user account with viewer privileges and then configure the event viewer to use this account to connect to the sensor.

- **Service**—This user role does not have direct access to the CLI. Service account users are logged directly into a bash shell. Use this account for support and troubleshooting purposes only. Unauthorized modifications are not supported and will require the device to be reimaged to guarantee proper operation. You can create only one user with the service role.

  When you log in to the service account, you receive the following warning:

```
****************************** WARNING ****************************************
UNAUTHORIZED ACCESS TO THIS NETWORK DEVICE IS PROHIBITED.
This account is intended to be used for support and troubleshooting purposes only.
Unauthorized modifications are not supported and will require this device to be
re-imaged to guarantee proper operation.
*******************************************************************************
```

Note The service role is a special role that allows you to bypass the CLI if needed. Only a user with Administrator privileges can edit the service account.

Service Account

The service account is a support and troubleshooting tool that enables TAC to log in to a native operating system shell rather than the CLI shell. It does not exist on the sensor by default. You must create it so that it available for TAC to use for troubleshooting your sensor. For the procedure to create the service account, see Creating the Service Account, page 4-13.

Only one service account is allowed per sensor and only one account is allowed a service role. When the service account's password is set or reset, the root account's password is set to the same password. This allows the service account user to su to root using the same password. When the service account is removed, the root account's password is locked.

The service account is not intended to be used for configuration purposes. Only modifications made to the sensor through the service account under the direction of TAC are supported. Cisco Systems does not support the addition and/or running of an additional service to the operating system through the service account, because it affects proper performance and proper functioning of the other IPS services. TAC does not support a sensor on which additional services have been added.

You can track logins to the service account by checking the log file /var/log/.tac, which is updated with a record of service account logins.

Note IPS 5.1 incorporates several troubleshooting features that are available through the CLI or IDM. The service account is not necessary for most troubleshooting situations. You may need to create the service account at the TAC's direction to troubleshoot a very unique problem. The service account lets you bypass the protections built into the CLI and allows root privilege access to the sensor, which is otherwise disabled. We recommend that you do not create a service account unless it is needed for a specific reason. You should remove the service account when it is no longer needed.

CLI Behavior

Follow these tips when using the IPS CLI:

Prompts

- You cannot change the prompt displayed for the CLI commands.
- User interactive prompts occur when the system displays a question and waits for user input. The default input is displayed inside brackets [ ]. To accept the default input, press **Enter**.

Help

- To display the help for a command, type **?** after the command.

  The following example demonstrates the ? function:

  ```
  sensor# configure ?
  terminal     Configure from the terminal
  sensor# configure
  ```

  Note When the prompt returns from displaying help, the command previously entered is displayed without the ?.

- You can type **?** after an incomplete token to view the valid tokens that complete the command. If there is a trailing space between the token and the ?, you receive an ambiguous command error:

  ```
  sensor# show c ?
  % Ambiguous command : "show c"
  ```

  If you enter the token without the space, a selection of available tokens for the completion (with no help description) appears:

  ```
  sensor# show c?
  clock configuration
  sensor# show c
  ```

- Only commands available in the current mode are displayed by help.

Tab Completion

- Only commands available in the current mode are displayed by tab complete and help.
- If you are unsure of the complete syntax for a command, you can type a portion of the command and press **Tab** to complete the command.
- If multiple commands match for tab completion, nothing is displayed.

Recall

- To recall the commands entered in a mode, use the Up Arrow or Down Arrow keys or press **Ctrl-P** or **Ctrl-N**.

  Note Help and tab complete requests are not reported in the recall list.

- A blank prompt indicates the end of the recall list.

Case Sensitivity

- The CLI is not case sensitive, but it does echo back the text in the same case you typed it. For example, if you type:

```
sensor# CONF
```

and press **Tab**, the sensor displays:

```
sensor# CONFigure
```

Display Options

- `–More–` is an interactive prompt that indicates that the terminal output exceeds the allotted display space. To display the remaining output, press the **spacebar** to display the next page of output or press **Enter** to display the output one line at a time.
- To clear the current line contents and return to a blank command line, press **Ctrl-C**.

Communications

This section describes the communications protocols used by IPS 5.1. It contains the following topics:

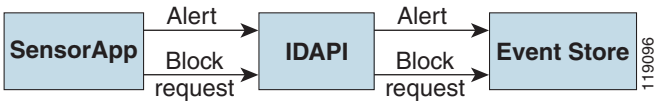
IDAPI

IPS applications use an interprocess communication API called IDAPI to handle internal communications. IDAPI reads and writes event data and provides a mechanism for control transactions. IDAPI is the interface through which all the applications communicate.

SensorApp captures and analyzes the network traffic on its interfaces. When a signature is matched, SensorApp generates an alert, which is stored in the Event Store. If the signature is configured to perform the blocking response action, SensorApp generates a block event, which is also stored in the Event Store.

Figure A-5 illustrates the IDAPI interface.

Figure A-5 IDAPI



Each application registers to the IDAPI to send and receive events and control transactions. IDAPI provides the following services:

- Control transactions
  - Initiates the control transaction.
  - Waits for the inbound control transaction.
  - Responds to the control transaction.
- IPS events
  - Subscribes to remote IPS events, which are stored in the Event Store when received.
  - Reads IPS events from the Event Store.
  - Writes IPS events to the Event Store.

IDAPI provides the necessary synchronization mechanisms to guarantee atomic data accesses.

RDEP2

External communications use RDEP2. RDEP2 is an application-level communications protocol used to exchange IPS event, IP log, configuration, and control messages between IPS clients and IPS servers. RDEP2 communications consist of request and response messages. RDEP2 clients initiate request messages to RDEP2 servers. RDEP2 servers respond to request messages with response messages.

RDEP2 defines three classes of request/response messages: event, IP log, and transaction messages. Event messages include IPS alert, status, and error messages. Clients use IP log requests to retrieve IP log data from servers. Transaction messages are used to configure and control IPS servers.

RDEP2 uses the industry standards HTTP, TLS and SSL and XML to provide a standardized interface between RDEP2 agents. The RDEP2 protocol is a subset of the HTTP 1.1 protocol. All RDEP2 messages are legal HTTP 1.1 messages. RDEP2 uses HTTP's message formats and message exchange protocol to exchange messages between RDEP2 agents.

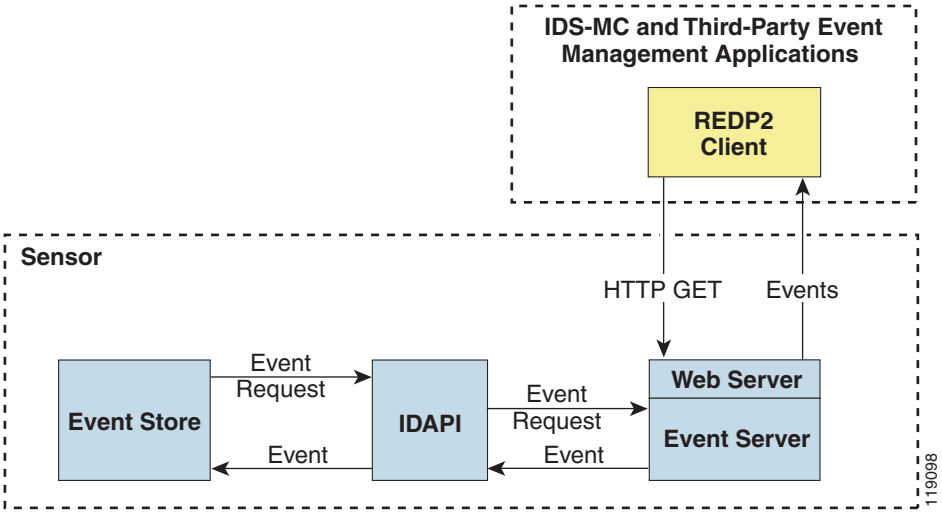
You use the IPS manager to specify which hosts are allowed to access the sensor through the network. Sensors accept connections from 1 to 10 RDEP2 clients simultaneously. Clients selectively retrieve data by time range, type of event (alert, error, or status message) and level (alert = high, medium, low, or informational; error = high, medium, low). Events are retrieved by a query (a single bulk get) or subscription (a real-time persistent connection) or both. Communications are secured by TLS or SSL.

Note For retrieving events, the sensor is backwards-compatible to RDEP even though the new standard for retrieval is RDEP2. We recommend you use RDEP2 to retrieve events and send configuration changes for IPS 5.1.

Remote applications retrieve events from the sensor through RDEP2. The remote client sends an RDEP2 event request to the sensor's Web Server, which passes it to the Event Server. The Event Server queries the Event Store through IDAPI and then returns the result.

Figure A-6 shows remote applications retrieving events from the sensor through RDEP2.

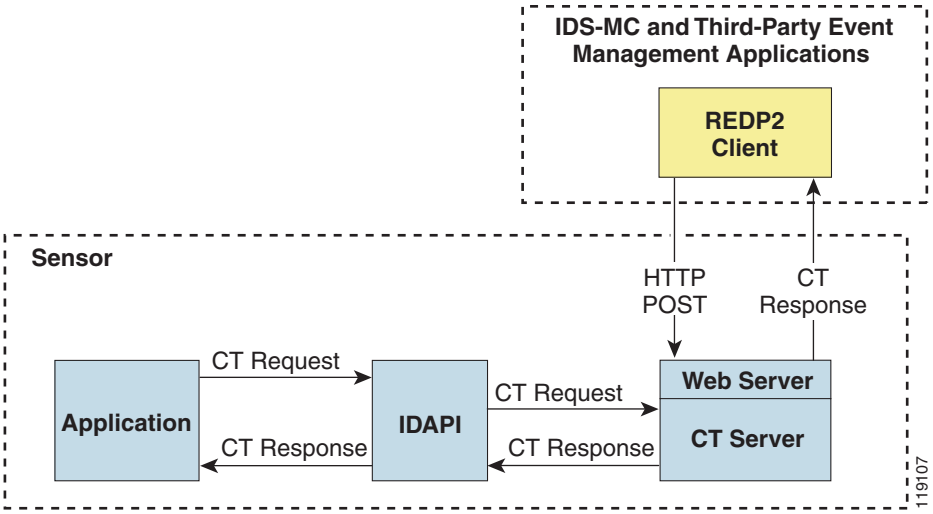
Figure A-6 Retrieving Events Through RDEP2



Remote applications send commands to the sensor through RDEP2. The remote client sends an RDEP2 control transaction to the sensor's Web Server, which passes it to the Control Transaction Server. The Control Transaction Server passes the control transaction through IDAPI to the appropriate application, waits for the application's response, and then returns the result.

Figure A-7 shows remote applications sending commands to the sensor through RDEP2.

Figure A-7 Sending Commands Through RDEP2



IDIOM

IDIOM is a data format standard that defines the event messages that are reported by the IPS as well as the operational messages that are used to configure and control intrusion detection systems. These messages consist of XML documents that conform to the IDIOM XML schema.

IDIOM supports two types of interactions: event and control transaction. Event interactions are used to exchange IPS events such as alerts. IDIOM uses two types of messages for event interactions: event and error messages. Control transactions provide a means for one host to initiate an action in, change the state of, or read the state of another host. Control transactions utilize four types of IDIOM messages: request, response, configuration, and error messages. Events and control transactions that are communicated between application instances within a host are known as local events or local control transactions, or collectively, local IDIOM messages. Events and control transactions that are communicated between different hosts using the RDEP2 protocol are known as remote events and remote control transactions, or collectively, remote IDIOM messages.

Note IDIOM for the most part has been superseded by IDCONF, SDEE, and CIDEE.

IDCONF

IPS 5.1 manages its configuration using XML documents. IDCONF specifies the XML schema including IPS 5.1 control transactions. The IDCONF schema does not specify the contents of the configuration documents, but rather the framework and building blocks from which the configuration documents are developed. It provides mechanisms that let the IPS managers and CLI ignore features that are not configurable by certain platforms or functions through the use of the feature-supported attribute.

IDCONF messages are exchanged over RDEP2 and are wrapped inside IDIOM request and response messages.

The following is an IDCONF example:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<request xmlns="http://www.cisco.com/cids/idiom" schemaVersion="2.00">
  <editConfigDelta xmlns="http://www.cisco.com/cids/idconf">
    <component name="userAccount">
      <config typedefsVersion="2004-03-01" xmlns="http://www.cisco.com/cids/idconf">
        <struct>
           <map name="user-accounts‟ editOp=“merge”>
             <mapEntry>
               <key>
                <var name="name">cisco</var>
              </key>
               <struct>
                <struct name="credentials">
                  <var name="role">administrator</var>
                </struct>
               </struct>
            </mapEntry>
          </map>
        </struct>
      </config>
    </component>
  </editDefaultConfig>
</request>
```

SDEE

IPS produces various types of events including intrusion alerts and status events. IPS communicates events to clients such as management applications using the proprietary RDEP2. We have also developed an IPS-industry leading protocol, SDEE, which is a product-independent standard for communicating security device events. SDEE is an enhancement to the current version of RDEP2 that adds extensibility features that are needed for communicating events generated by various types of security devices.

Systems that use SDEE to communicate events to clients are referred to as SDEE providers. SDEE specifies that events can be transported using the HTTP or HTTP over SSL and TLS protocols. When HTTP or HTTPS is used, SDEE providers act as HTTP servers, while SDEE clients are the initiators of HTTP requests.

IPS includes Web Server, which processes HTTP or HTTPS requests. Web Server uses run-time loadable servlets to process the different types of HTTP requests. Each servlet handles HTTP requests that are directed to the URL associated with the servlet. The SDEE server is implemented as a web server servlet.

The SDEE server only processes authorized requests. A request is authorized if is originates from a web server to authenticate the client's identity and determine the client's privilege level.

CIDEE

CIDEE specifies the extensions to SDEE that are used by the Cisco IPS. The CIDEE standard specifies all possible extensions that are supported by IPS. Specific systems may implement a subset of CIDEE extensions. However, any extension that is designated as being required MUST be supported by all systems.

CIDEE specifies the IPS-specific security device events as well as the IPS extensions to the SDEE evIdsAlert element.

CIDEE supports the following events:

- evError—Error event

  Generated by the CIDEE provider when the provider detects an error or warning condition. The evError event contains error code and textual description of the error.

- evStatus—Status message event

  Generated by CIDEE providers to indicate that something of potential interest occurred on the host. Different types of status messages can be reported in the status event—one message per event. Each type of status message contains a set of data elements that are specific to the type of occurrence that the status message is describing. The information in many of the status messages may be useful for audit purposes. Errors and warnings are not considered status information and are reported using evError rather than evStatus.

- evShunRqst—Block request event

  Generated to indicate that a block action is to be initiated by the service that handles network blocking.

The following is a CDIEE extended event example:

```
<sd:events xmlns:cid="http://www.cisco.com/cids/2004/04/cidee"
xmlns:sd="“http://example.org/2003/08/sdee”>
 <sd:evIdsAlert eventId="1042648730045587005" vendor="Cisco‟ severity="medium">
  <sd:originator>
    <sd:hostId>Beta4Sensor1</sd:hostId>
    <cid:appName>sensorApp</cid:appName>
    <cid:appInstanceId>8971</cid:appInstanceId>
```

```
</sd:originator>
<sd:time offset="0" timeZone="UTC">1043238671706378000</sd:time>
<sd:signature description="IOS Udp Bomb" id="4600" cid:version="S37">
  <cid:subsigId>0</cid:subsigId>
</sd:signature>
…
```

IPS 5.1 File Structure

IPS 5.1 has the following directory structure:

- /usr/cids/idsRoot—Main installation directory.
- /usr/cids/idsRoot/shared—Stores files used during system recovery.
- /usr/cids/idsRoot/var—Stores files created dynamically while the sensor is running.
- /usr/cids/idsRoot/var/updates—Stores files and logs for update installations.
- /usr/cids/idsRoot/var/virtualSensor—Stores files used by SensorApp to analyze regular expressions.
- /usr/cids/idsRoot/var/eventStore—Contains the Event Store application.
- /usr/cids/idsRoot/var/core—Stores core files that are created during system crashes.
- /usr/cids/idsRoot/var/iplogs—Stores iplog file data.
- /usr/cids/idsRoot/bin—Contains the binary executables.
- /usr/cids/idsRoot/bin/authentication—Contains the authentication application.
- /usr/cids/idsRoot/bin/cidDump—Contains the script that gathers data for tech support.
- /usr/cids/idsRoot/bin/cidwebserver—Contains the web server application.
- /usr/cids/idsRoot/bin/cidcli—Contains the CLI application.
- /usr/cids/idsRoot/bin/nac—Contains the ARC application.
- /usr/cids/idsRoot/bin/logApp—Contains the logger application.
- /usr/cids/idsRoot/bin/mainApp—Contains the main application.
- /usr/cids/idsRoot/bin/sensorApp—Contains the sensor application.
- /usr/cids/idsRoot/bin/falcondump—Contains the application for getting packet dumps on the sensing ports of the IDS-4250-XL and IDSM-2.
- /usr/cids/idsRoot/etc—Stores sensor configuration files.
- /usr/cids/idsRoot/htdocs—Contains the IDM files for the web server.
- /usr/cids/idsRoot/lib—Contains the library files for the sensor applications.
- /usr/cids/idsRoot/log—Contains the log files for debugging.
- /usr/cids/idsRoot/tmp—Stores the temporary files created during run time of the sensor.

Summary of IPS 5.1 Applications

Table A-2 gives a summary of the applications that make up the IPS.

Table A-2 Summary of Applications

Application	Description
AuthenticationApp	Authorizes and authenticates users based on IP address, password, and digital certificates.
CLI	Accepts command line input and modifies the local configuration using IDAPI.
Event Server[1]	Accepts RDEP2 request for events from remote clients.
MainApp	Reads the configuration and starts applications, handles starting and stopping of applications and node reboots, handles software upgrades.
InterfaceApp	Handles bypass and physical settings and defines paired interfaces. Physical settings are speed, duplex, and administrative state.
LogApp	Writes all the application's log messages to the log file and the application's error messages to the Event Store.
Attack Response Controller	An ARC is run on every sensor. Each ARC subscribes to network access events from its local Event Store. The ARC configuration contains a list of sensors and the network access devices that its local ARC controls. If a ARC is configured to send network access events to a master blocking sensor, it initiates a network access control transaction to the remote ARC that controls the device. These network access action control transactions are also used by IPS managers to issue occasional network access actions.
NotificationApp	Sends SNMP traps when triggered by alert, status, and error events. NotificationApp uses the public domain SNMP agent. SNMP GETs provide information about the general health of the sensor.
SensorApp	Captures and analyzes traffic on the monitored network and generates intrusion and network access events. Responds to IP logging control transactions that turn logging on and off and that send and delete IP log files.
Control Transaction Server[2]	Accepts control transactions from a remote RDEP2 client, initiates a local control transaction, and returns the response to the remote client.
Control Transaction Source[3]	Waits for control transactions directed to remote applications, forwards the control transactions to the remote node using RDEP2, and returns the response to the initiator.
IDM	The Java applet that provides an HTML IPS management interface.
Web Server	Waits for remote HTTP client requests and calls the appropriate servlet application.

1. This is a web server servlet.
2. This is a web server servlet.
3. This is a remote control transaction proxy.

APPENDIX **B**

Signature Engines

This appendix describes the IPS signature engines. It contains the following sections:

About Signature Engines

A signature engine is a component of the Cisco IPS that is designed to support many signatures in a certain category. An engine is composed of a parser and an inspector. Each engine has a set of parameters that have allowable ranges or sets of values.

Note The 5.1 engines support a standardized Regex.

IPS 5.1 contains the following signature engines:

- AIC—Provides thorough analysis of web traffic.

  It provides granular control over HTTP sessions to prevent abuse of the HTTP protocol. It allows administrative control over applications, such as instant messaging and gotomypc, that try to tunnel over specified ports. You can also use AIC to inspect FTP traffic and control the commands being issued.

  There are two AIC engines: AIC FTP and AIC HTTP.

  For more information on configuring the AIC engine signatures, see Configuring AIC Signatures, page 7-13.

- Atomic—The Atomic engines are now combined into two engines with multi-level selections. You can combine Layer-3 and Layer-4 attributes within one signature, for example IP + TCP. The Atomic engine uses the standardized Regex support.

  - Atomic IP —Inspects IP protocol packets and associated Layer-4 transport protocols.

    This engine lets you specify values to match for fields in the IP and Layer-4 headers, and lets you use Regex to inspect Layer-4 payloads.

    Note All IP packets are inspected by the Atomic IP engine. This engine replaces the 4.x Atomic ICMP, Atomic IP Options, Atomic L3 IP, Atomic TCP, and Atomic UDP engines.

  - Atomic ARP—Inspects Layer-2 ARP protocol. The Atomic ARP engine is different because most engines are based on Layer-3-IP.

- Flood—Detects ICMP and UDP floods directed at hosts and networks.

  There are two Flood engines: Flood HOST and Flood NET.

- Meta—Defines events that occur in a related manner within a sliding time interval. This engine processes events rather than packets.

- Multi String—Inspects Layer 4 transport protocols and payloads by matching several strings for one signature.

  This engine inspects stream-based TCP and single UDP and ICMP packets.

  Note The Multi String engine is new for IPS 5.1.

- Normalizer—Configures how the IP and TCP normalizer functions and provides configuration for signature events related to the IP and TCP normalizer. Allows you to enforce RFC compliance.

- Service—Deals with specific protocols. Service engine has the following protocol types:

  - DNS—Inspects DNS (TCP and UDP) traffic.
  - FTP—Inspects FTP traffic.
  - GENERIC—Decodes custom service and payload.
  - H225— Inspects VoIP traffic.

    Helps the network administrator make sure the SETUP message coming in to the VoIP network is valid and within the bounds that the policies describe. Is also helps make sure the addresses and Q.931 string fields such as url-ids, email-ids, and display information adhere to specific lengths and do not contain possible attack patterns.

  - HTTP—Inspects HTTP traffic.

    The WEBPORTS variable defines inspection port for HTTP traffic.
  - IDENT—Inspects IDENT (client and server) traffic.
  - MSRPC—Inspects MSRPC traffic.
  - MSSQL—Inspects Microsoft SQL traffic.
  - NTP—Inspects NTP traffic.
  - RPC—Inspects RPC traffic.
  - SMB—Inspects SMB traffic.
  - SNMP—Inspects SNMP traffic.
  - SSH—Inspects SSH traffic.
- State—Stateful searches of strings in protocols such as SMTP.

  The state engine now has a hidden configuration file that is used to define the state transitions so new state definitions can be delivered in a signature update.
- String—Searches on Regex strings based on ICMP, TCP, or UDP protocol.

  There are three String engines: String ICMP, String TCP, and String UDP.
- Sweep—Analyzes sweeps from a single host (ICMP and TCP), from destination ports (TCP and UDP), and multiple ports with RPC requests between two nodes.
- Traffic ICMP—Analyzes nonstandard protocols, such as TFN2K, LOKI, and DDOS. There are only two signatures with configurable parameters.
- Trojan—Analyzes traffic from nonstandard protocols, such as BO2K andTFN2K.

  There are three Trojan engines: Bo2k, Tfn2k, and UDP. There are no user-configurable parameters in these engines.

Master Engine

The Master engine provides structures and methods to the other engines and handles input from configuration and alert output. This section describes the Master engine, and contains the following topics:

General Parameters

The following parameters are part of the Master engine and apply to all signatures.

Table B-1 lists the general master engine parameters.

Table B-1 Master Engine General Parameters

Parameter	Description	Value
alert-severity	Severity of the alert: • Dangerous alert • Medium-level alert • Low-level alert • Informational alert	high medium low informational
engine	Specifies the engine the signature belongs to.	—
event-counter	Grouping for event count settings.	—
event-count	Number of times an event must occur before an alert is generated.	1 to 65535
event-count-key	The storage type on which to count events for this signature: • Attacker address • Attacker and victim addresses • Attacker address and victim port • Victim address • Attacker and victim addresses and ports	Axxx AxBx Axxb xxBx AaBb
specify-alert-interval	Enables alert interval.	yes | no
alert-interval	Time in seconds before the event count is reset.	2 to 1000
promisc-delta	Delta value used to determine seriousness of the alert.	0 to 30
sig-fidelity-rating	Rating of the fidelity of this signature.	0 to 100
sig-description	Grouping for your description of the signature.	—
sig-name	Name of the signature.	*sig-name*
sig-string-info	Additional information about this signature that will be included in the alert message.	*sig-string-info*
sig-comment	Comments about this signature.	*sig-comment*
alert-traits	Traits you want to document about this signature.	0 to 65335
release	The release in which the signature was most recently updated.	*release*
status	Whether the signature is enabled or disabled, active or retired.	enabled retired

Caution We do not recommend that you change the promisc-delta setting for a signature.

Promiscuous delta lowers the RR of certain alerts in promiscuous mode. Because the sensor does not know the attributes of the target system and in promiscuous mode cannot deny packets, it is useful to lower the prioritization of promiscuous alerts (based on the lower risk rating) so the administrator can focus on investigating higher risk rating alerts.

In inline mode, the sensor can deny the offending packets and they never reach the target host, so it does not matter if the target was vulnerable. The attack was not allowed on the network and so we do not subtract from the risk rating value.

Signatures that are not service, OS, or application-specific have 0 for the promiscuously delta. If the signature is specific to an OS, service, or application, it has a promiscuous delta of 5, 10, or 15 calculated from 5 points for each category.

Alert Frequency

The purpose of the alert frequency parameter is to reduce the volume of the alerts written to the Event Store to counter IDS DoS tools, such as stick. There are four modes: Fire All, Fire Once, Summarize, and Global Summarize. The summary mode is changed dynamically to adapt to the current alert volume. For example, you can configure the signature to Fire All, but after a certain threshold is reached, it starts summarizing.

Table B-2 lists the alert frequency parameters.

Table B-2 Master Engine Alert Frequency Parameters

Parameter	Description	Value
alert-frequency	Summary options for grouping alerts.	—
summary-mode	Mode used for summarization.	—
fire-all	Fires an alert on all events.	—
fire-once	Fires an alert only once.	—
global-summarize	Summarizes an alert so that it only fires once regardless of how many attackers or victims.	—
summarize	Summarizes alerts.	—
specify-summary-threshold	(Optional) Enables summary threshold.	yes | no
summary-threshold	Threshold number of alerts to send signature into summary mode.	0 to 65535
specify-global-summary-threshold	Enable global summary threshold.	yes | no
global-summary-threshold	Threshold number of events to take alerts into global summary.	1 to 65535
summary-interval	Time in seconds used in each summary alert.	1 to 1000
summary-key	The storage type on which to summarize this signature: • Attacker address • Attacker and victim addresses • Attacker address and victim port • Victim address • Attacker and victim addresses and ports	Axxx AxBx Axxb xxBx AaBb

Event Actions

Most of the following event actions belong to each signature engine unless they are not appropriate for that particular engine.

Table B-3 describes the event actions.

Table B-3 Event Actions

Event Action Name	Description
Deny Attacker Inline	(Inline mode only) Does not transmit this packet and future packets originating from the attacker address for a specified period of time.[1] **Note** This is the most severe of the deny actions. It denies current and future packets from a single attacker address. To clear all denied attacker entries, choose **Monitoring > Denied Attackers > Clear List**, which permits the addresses back on the network. For the procedure, see Monitoring and Clearing the Denied Attackers List, page 6-21.
Deny Attacker Service Pair Inline	(Inline mode only) Does not transmit this packet and future packets on the attacker address victim port pair for a specified period of time.
Deny Attacker Victim Pair Inline	(Inline mode only) Does not transmit this packet and future packets on the attacker/victim address pair for a specified period of time. **Note** For deny actions, to set the specified period of time and maximum number of denied attackers choose **Configuration > Event Action Rules > General Settings**. For the procedure, see Configuring the General Settings, page 6-20.
Deny Connection Inline	(Inline mode only) Does not transmit this packet and future packets on the TCP flow.
Deny Packet Inline	(Inline mode only) Does not transmit this packet. **Note** You cannot delete the event action override for Deny Packet Inline because it is protected. If you do not want to use that override, disable it.
Log Attacker Packets	Starts IP logging packets containing the attacker address. **Note** This action causes an alert to be written to the Event Store, even if Produce Alert is not selected.
Log Pair Packets	Starts IP logging packets containing the attacker-victim address pair. **Note** This action causes an alert to be written to the Event Store, even if Produce Alert is not selected.
Log Victim Packets	Starts IP logging packets containing the victim address.
Modify Packet Inline	Modifies packet data to remove ambiguity about what the end point might do with the packet. **Note** Modify Packet Inline is not an option for Add Event Action Filter or Add Event Action Override.

Table B-3 Event Actions (continued)

Event Action Name	Description
Produce Alert	Writes the event to the Event Store as an alert. **Note** The Produce Alert action is not automatic when you enable alerts for a signature. To have an alert created in the Event Store, you must select Produce Alert. If you add a second action, you must include Produce Alert if you want an alert sent to the Event Store. Also, every time you configure the event actions, a new list is created and it replaces the old list. Make sure you include all the event actions you need for each signature.
Produce Verbose Alert	Includes an encoded dump of the offending packet in the alert. **Note** This action causes an alert to be written to the Event Store, even if Produce Alert is not selected.
Request Block Connection	Sends a request to ARC to block this connection. **Note** You must have blocking devices configured to implement this action. For more information, see Chapter 10, "Configuring Attack Response Controller for Blocking and Rate Limiting."
Request Block Host	Sends a request to ARC to block this attacker host. **Note** You must have blocking devices configured to implement this action. For more information, see Chapter 10, "Configuring Attack Response Controller for Blocking and Rate Limiting." **Note** For block actions, to set the duration of the block choose **Configuration >Event Action Rules > General Settings**. For the procedure, see Chapter 6, "Configuring the General Settings."
Request Rate Limit	Sends a rate limit request to ARC to perform rate limiting. You must have rate limiting devices configured to implement this action. For more information, see Chapter 10, "Configuring Attack Response Controller for Blocking and Rate Limiting."
Request SNMP Trap	Sends a request to NotificationApp to perform SNMP notification. **Note** This action causes an alert to be written to the Event Store, even if Produce Alert is not selected. You must have SNMP configured on the sensor to implement this action. For more information, see Chapter 11, "Configuring SNMP."
Reset TCP Connection	Sends TCP resets to hijack and terminate the TCP flow. **Note** Reset TCP Connection only works on TCP signatures that analyze a single connection. It does not work for sweeps or floods.

1. The sensor maintains a list of attackers being denied by the system. To remove an entry from the denied attacker list, you can view the list of attackers and clear the entire list, or you can wait for the timer to expire. The timer is a sliding timer for each entry. Therefore, if attacker A is being denied, but issues another attack, the timer for attacker A is reset and attacker A remains in the denied attacker list until the timer expires. If the denied attacker list is at capacity and cannot add a new entry, the packet will still be denied.

Understanding Deny Packet Inline

For signatures that have deny-packet-inline configured as an action or for an event action override that adds deny-packet-inline as an action, the following actions may be taken:

- droppedPacket
- deniedFlow
- tcpOneWayResetSent

The deny packet inline action is represented as a dropped packet action in the alert. When a deny packet inline occurs for a TCP connection, it is automatically upgraded to a deny connection inline action and seen as a denied flow in the alert. If the IPS denies just one packet, the TCP continues to try to send that same packet again and again, so the IPS denies the entire connection to ensure it never succeeds with the resends.

When a deny connection inline occurs, the IPS also automatically sends a TCP one-way reset, which shows up as a TCP one-way reset sent in the alert. When the IPS denies the connection, it leaves an open connection on both the client (generally the attacker) and the server (generally the victim). Too many open connections can result in resource problems on the victim. So the IPS sends a TCP reset to the victim to close the connection on the victim side (usually the server), which conserves the resources of the victim. It also prevents a failover that would otherwise allow the connection to fail over to a different network path and reach the victim. The IPS leaves the attacker side open and denies all traffic from it.

AIC Engine

The AIC engine inspects HTTP web traffic and enforces FTP commands. This section describes the AIC engine and its parameters, and contains the following topics:

Overview

The AIC engine defines signatures for deep inspection of web traffic. It also defines signatures that authorize and enforce FTP commands.

There are two AIC engines: AIC HTTP and AIC FTP.

The AIC engine has the following features:

- Web traffic:
  - RFC compliance enforcement
  - HTTP request method authorization and enforcement
  - Response message validation
  - MIME type enforcement
  - Transfer encoding type validation
  - Content control based on message content and type of data being transferred
  - URI length enforcement
  - Message size enforcement according to policy configured and the header

  - Tunneling, P2P and instant messaging enforcement.

    This enforcement is done using regular expressions. There are predefined signature but you can expand the list.

- FTP traffic:
  - FTP command authorization and enforcement

AIC Engine Parameters

AIC provides thorough analysis of web traffic. It provides granular control over HTTP sessions to prevent abuse of the HTTP protocol. It allows administrative control over applications, such as instant messaging and gotomypc, that try to tunnel over specified ports. Inspection and policy checks for P2P and instant messaging are possible if these applications are running over HTTP.

AIC also provides a way to inspect FTP traffic and control the commands being issued.

You can enable or disable the predefined signatures or you can create policies through custom signatures.

The AIC engine runs when HTTP traffic is received on AIC web ports. If traffic is web traffic, but not received on the AIC web ports, the Service HTTP engine is executed. AIC inspection can be on any port if it is configured as an AIC web port and the traffic to be inspected is HTTP traffic.

Caution The AIC web ports are regular HTTP web ports. You can turn on AIC web ports to distinguish which ports should watch for regular HTTP traffic and which ports should watch for AIC enforcement. You might use AIC web ports, for example, if you have a proxy on port 82 and you need to monitor it. We recommend that you do not configure separate ports for AIC enforcement.

For the procedures for configuring AIC engine signatures, see Configuring AIC Signatures, page 7-13. For an example of a custom AIC signature, see Example AIC MIME-Type Signature, page 7-42.

Table B-4 lists the parameters that are specific to the AIC HTTP engine.

Table B-4 AIC HTTP Engine Parameters

Parameter	Description
signature-type	Specifies the type of AIC signature.
content-types	AIC signature that deals with MIME types: • define-content-type associates actions such as denying a specific MIME type (image/gif), defining a message-size violation, and determining that the MIME-type mentioned in the header and body do not match. • define-recognized-content-types lists content types recognized by the sensor.
define-web-traffic-policy	Specifies the action to take when noncompliant HTTP traffic is seen. The **alarm-on-non-http-traffic [true | false]** command enables the signature. This signature is disabled by default.
max-outstanding-requests-overrun	Maximum allowed HTTP requests per connection (1 to 16).
msg-body-pattern	Uses Regex to define signatures that look for specific patterns in the message body.

Table B-4 AIC HTTP Engine Parameters (continued)

Parameter	Description
request-methods	AIC signature that allows actions to be associated with HTTP request methods: • define-request-method, such as get, put, and so forth. • recognized-request-methods lists methods recognized by the sensor.
transfer-encodings	AIC signature that deals with transfer encodings: • define-transfer-encoding associates an action with each method, such as compress, chunked, and so forth. • recognized-transfer-encodings lists methods recognized by the sensor. • chunked-transfer-encoding-error specifies actions to be taken when a chunked encoding error is seen.

Table B-5 lists the parameters that are specific to the AIC FTP engine.

Table B-5 AIC FTP Engine Parameters

Parameter	Description
signature-type	Specifies the type of AIC signature.
ftp-commands	Associates an action with an FTP command: • ftp-command—Lets you choose the FTP command you want to inspect.
unrecognized-ftp-command	Inspects unrecognized FTP commands.

Atomic Engine

The Atomic engine contains signatures for simple, single packet conditions that cause alerts to be fired. This section describes the Atomic engine, and contains the following topics:

- Atomic ARP Engine, page B-11
- Atomic IP Engine, page B-11

Atomic ARP Engine

The Atomic ARP engine defines basic Layer-2 ARP signatures and provides more advanced detection of the ARP spoof tools dsniff and ettercap.

Table B-6 lists the parameters that are specific to the Atomic ARP engine.

Table B-6 Atomic ARP Engine Parameters

Parameter	Description
specify-mac-flip	Fires an alert when the MAC address changes more than this many times for this IP address.
specify-type-of-arp-sig	Specifies the type of ARP signatures you want to fire on: • Source Broadcast (default)—Fires an alert for this signature when it sees an ARP source address of 255.255.255.255. • Destination Broadcast—Fires an alert for this signature when it sees an ARP destination address of 255.255.255.255. • Same Source and Destination—Fires an alert for this signature when it sees an ARP destination address with the same source and destination MAC address • Source Multicast—Fires an alert for this signature when it sees an ARP source MAC address of 01:00:5e:(00-7f).
specify-request-inbalance	Fires an alert when there are this many more requests than replies on the IP address.
specify-arp-operation	The ARP operation code for this signature.

Atomic IP Engine

The Atomic IP engine defines signatures that inspect IP protocol headers and associated Layer-4 transport protocols (TCP, UDP, and ICMP) and payloads.

Note The Atomic engines do not store persistent data across packets. Instead they can fire an alert from the analysis of a single packet.

Table B-7 lists the parameters that are specific to the Atomic IP engine.

Table B-7 Atomic IP Engine Parameters

Parameter	Description
fragment-status	Specifies whether or not fragments are wanted.
specify-ip-payload-length	Specifies IP datagram payload length.
specify-ip-header-length	Specifies IP datagram header length.
specify-ip-addr-options	Specifies IP addresses.
specify-ip-id	Specifies IP identifier.
specify-ip-total-length	Specifies IP datagram total length.

Table B-7 Atomic IP Engine Parameters (continued)

Parameter	Description
specify-ip-option-inspection	Specifies IP options inspection.
specify-l4-protocol	Specifies Layer 4 protocol.
specify-ip-tos	Specifies type of server.
specify-ip-ttl	Specifies time to live.
specify-ip-version	Specifies IP protocol version.

Flood Engine

The Flood engine defines signatures that watch for any host or network sending multiple packets to a single host or network. For example, you can create a signature that fires when 150 or more packets per second (of the specific type) are found going to the victim host. There are two types of Flood engines: Flood Host and Flood Net.

Table B-8 lists the parameters specific to the Flood Host engine.

Table B-8 Flood Host Engine Parameters

Parameter	Description	Value
protocol	Which kind of traffic to inspect.	ICMP UDP
rate	Threshold number of packets per second.	0 to 65535[1]
icmp-type	Specifies the value for the ICMP header type.	0 to 65535
dst-ports	Specifies the destination ports when you choose UDP protocol.	0 to 65535[2] a-b[,c-d]
src-ports	Specifies the source ports when you choose UDP protocol.	0 to 65535[3] a-b[,c-d]

1. An alert fires when the rate is greater than the packets per second.
2. The second number in the range must be greater than or equal to the first number.
3. The second number in the range must be greater than or equal to the first number.

Table B-9 lists the parameters specific to the Flood Net engine.

Table B-9 Flood Net Engine Parameters

Parameter	Description	Value
gap	Gap of time allowed (in seconds) for a flood signature.	0 to 65535
peaks	Number of allowed peaks of flood traffic.	0 to 65535
protocol	Which kind of traffic to inspect.	ICMP TCP UDP
rate	Threshold number of packets per second.	0 to 65535[1]

Table B-9 Flood Net Engine Parameters (continued)

Parameter	Description	Value
sampling-interval	Interval used for sampling traffic.	1 to 3600
icmp-type	Specifies the value for the ICMP header type.	0 to 65535

1. An alert fires when the rate is greater than the packets per second.

Meta Engine

The Meta engine defines events that occur in a related manner within a sliding time interval. This engine processes events rather than packets. As signature events are generated, the Meta engine inspects them to determine if they match any or several Meta definitions. The Meta engine generates a signature event after all requirements for the event are met.

All signature events are handed off to the Meta engine by SEAP. SEAP hands off the event after processing the minimum hits option. Summarization and event action are processed after the Meta engine has processed the component events. For more information about SEAP, see Signature Event Action Processor, page 6-2.

Caution A large number of Meta signatures could adversely affect overall sensor performance.

Table B-10 lists the parameters specific to the Meta engine.

Table B-10 Meta Engine Parameters

Parameter	Description	Value
meta-reset-interval	Time in seconds to reset the META signature.	0 to 3600
component-list	List of Meta components: • edit—Edits an existing entry • insert—Inserts a new entry into the list: – begin—Places the entry at the beginning of the active list – end—Places the entry at the end of the active list – inactive—Places the entry into the inactive list – before—Places the entry before the specified entry – after—Places the entry after the specified entry • move—Moves an entry in the list	*name1*
meta-key	Storage type for the Meta signature: • Attacker address • Attacker and victim addresses • Attacker and victim addresses and ports • Victim address	AaBb AxBx Axxx xxBx

Table B-10 Meta Engine Parameters (continued)

Parameter	Description	Value
unique-victim-ports	Number of unique victims ports required per Meta signature.	1 to 256
component-list-in-order	Whether to fire the component list in order.	true | false

For an example of a custom Meta engine signature, see Example MEG Signature, page 7-39.

Multi String Engine

The Multi String engine lets you define signatures that inspect Layer 4 transport protocol (ICMP, TCP, and UDP) payloads using multiple string matches for one signature. You can specify a series of regular expression patterns that must be matched to fire the signature. For example, you can define a signature that looks for regex 1 followed by regex 2 on a UDP service. For UDP and TCP you can specify port numbers and direction. You can specify a single source port, a single destination port, or both ports. The string matching takes place in both directions.

Use the Multi String engine when you need to specify more than one regex pattern. Otherwise, you can use the String ICMP, String TCP, or String UDP engine to specify a single regex pattern for one of those protocols.

Table B-11 lists the parameters specific to the Multi String Engine.

Table B-11 Multi String Engine Parameters

Parameter	Description	Value
inspect-length	Length of stream or packet that must contain all offending strings for the signature to fire.	0 to 4294967295
protocol	Layer 4 protocol selection.	icmp tcp udp
regex-component	List of regex components: • regex-string—The string to search for. • spacing-type—Type of spacing required from the match before or from the beginning of the stream/packet if it is the first entry in the list.	list (1 to 16 items) exact minimum
port-selection	Type of TCP or UDP port to inspect: • both-ports—Specifies both source and destination port. • dest-ports—Specifies a range of destination ports. • source-ports—Specifies a range of source ports.[1]	0 to 65535 [2]

Table B-11 Multi String Engine Parameters (continued)

Parameter	Description	Value
exact-spacing	Exact number of bytes that must be between this regex string and the one before, or from the beginning of the stream/packet if it is the first entry in the list.	0 to 4294967296
min-spacing	Minimum number of bytes that must be between this regex string and the one before, or from the beginning of the stream/packet if it is the first entry in the list.	0 to 4294967296
swap-attacker-victim	Swaps the attacker and victim addresses and ports (source and destination) in the alert message and in any actions taken.	true | false (default)

1. Port matching is performed bidirectionally for both the client-to-server and server-to-client traffic flow directions. For example, if the source-ports value is 80, in a client-to-server traffic flow direction, inspection occurs if the client port is 80. In a server-to-client traffic flow direction, inspection occurs if the server port is port 80.
2. A valid value is a comma- separated list of integer ranges a-b[,c-d] within 0 to 65535. The second number in the range must be greater than or equal to the first number.

Caution The Multi String engine can have a significant impact on memory usage.

Normalizer Engine

The Normalizer engine deals with IP fragmentation and TCP normalization. This section describes the Normalizer engine, and contains the following topics:

- Overview, page B-15
- Normalizer Engine Parameters, page B-16

Overview

The Normalizer engine deals with IP fragment reassembly and TCP stream reassembly. With the Normalizer engine you can set limits on system resource usage, for example, the maximum number of fragments the sensor tries to track at the same time.

Note You cannot add custom signatures to the Normalizer engine. You can tune the existing ones.

- IP Fragmentation Normalization

  Intentional or unintentional fragmentation of IP datagrams can hide exploits making them difficult or impossible to detect. Fragmentation can also be used to circumvent access control policies like those found on firewalls and routers. And different operating systems use different methods to queue and dispatch fragmented datagrams. If the sensor has to check for all possible ways that the end host will reassemble the datagrams, the sensor becomes vulnerable to denial of service attacks.

Reassembling all fragmented datagrams inline and only forwarding completed datagrams, refragmenting the datagram if necessary, prevents this. The IP Fragmentation Normalization unit performs this function.

- TCP Normalization

  Through intentional or natural TCP session segmentation, some classes of attacks can be hidden. To make sure policy enforcement can occur with no false positives and false negatives, the state of the two TCP endpoints must be tracked and only the data that is actually processed by the real host endpoints should be passed on. Overlaps in a TCP stream can occur, but are extremely rare except for TCP segment retransmits. Overwrites in the TCP session should not occur. If overwrites do occur, someone is intentionally trying to elude the security policy or the TCP stack implementation is broken. Maintaining full information about the state of both endpoints is not possible unless the sensor acts as a TCP proxy. Instead of the sensor acting as a TCP proxy, the segments will be ordered properly and the normalizer will look for any abnormal packets associated with evasion and attacks.

Sensors in promiscuous mode report alerts on violations. Sensors in inline mode perform the action specified in the event-action parameter, such as produce-alert, deny-packet-inline, and modify-packet-inline.

For the procedures for configuring signatures in the Normalizer engine, see Configuring IP Fragment Reassembly, page 7-21, and Configuring TCP Stream Reassembly, page 7-24.

Normalizer Engine Parameters

Table B-12 lists the parameters that are specific to the Normalizer engine.

Table B-12 Normalizer Engine Parameters

Parameter	Description
edit-default-sigs-only	Editable signatures.
specify-fragment-reassembly-timeout	(Optional) Enables fragment reassembly timeout.
specify-hijack-max-old-ack	(Optional) Enables hijack-max-old-ack.
specify-max-dgram-size	(Optional) Enables maximum datagram size.
specify-max-fragments	(Optional) Enables maximum fragments.
specify-max-fragments-per-dgram	(Optional) Enables maximum fragments per datagram.
specify-max-last-fragments	(Optional) Enables maximum last fragments.
specify-max-partial-dgrams	(Optional) Enables maximum partial datagrams.
specify-max-small-frags	(Optional) Enables maximum small fragments.
specify-min-fragment-size	(Optional) Enables minimum fragment size.
specify-service-ports	(Optional) Enables service ports.
specify-syn-flood-max-embryonic	(Optional) Enables SYN flood maximum embryonic.
specify-tcp-closed-timeout	(Optional) Enables TCP closed timeout.
specify-tcp-embryonic-timeout	(Optional) Enables TCP embryonic timeout.
specify-tcp-idle-timeout	(Optional) Enables TCP idle timeout.
specify-tcp-max-mss	(Optional) Enables TCP maximum mss.
specify-tcp-max-queue	(Optional) Enables TCP maximum queue.

Table B-12 Normalizer Engine Parameters (continued)

Parameter	Description
specify-tcp-min-mss	(Optional) Enables TCP minimum mss.
specify-tcp-option-number	(Optional) Enables TCP option number.

Service Engines

The Service engines analyze L5+ traffic between two hosts. These are one-to-one signatures that track persistent data. The engines analyze the L5+ payload in a manner similar to the live service.

The Service engines have common characteristics but each engine has specific knowledge of the service that it is inspecting. The Service engines supplement the capabilities of the generic string engine specializing in algorithms where using the string engine is inadequate or undesirable.

This section contains the following topics:

Service DNS Engine

The Service DNS engine specializes in advanced DNS decode, which includes anti-evasive techniques, such as following multiple jumps. It has many parameters such as lengths, opcodes, strings, and so forth. The Service DNS engine is a biprotocol inspector operating on both TCP and UDP port 53. It uses the stream for TCP and the quad for UDP.

Table B-13 lists the parameters specific to the Service DNS engine.

Table B-13 Service DNS Engine Parameters

Parameter	Description	Value
protocol	Protocol of interest for this inspector.	TCP UDP
specify-query-chaos-string	(Optional) Enables the DNS Query Class Chaos String.	*query-chaos-string*
specify-query-class	(Optional) Enables the query class: • query-class—DNS Query Class 2 Byte Value	0 to 65535
specify-query-invalid-domain-name	(Optional) Enables query invalid domain name: • query-invalid-domain-name—DNS Query Length greater than 255	true | false
specify-query-jump-count-exceeded	(Optional) Enables query jump count exceeded: • query-jump-count-exceeded—DNS compression counter	true | false
specify-query-opcode	(Optional) Enables query opcode: • query-opcode—DNS Query Opcode 1 byte Value	0 to 65535
specify-query-record-data-invalid	(Optional) Enables query record data invalid: • query-record-data-invalid—DNS Record Data incomplete	true | false
specify-query-record-data-len	(Optional) Enables the query record data length: • query-record-data-len—DNS Response Record Data Length	0 to 65535
specify-query-src-port-53	(Optional) Enables the query source port 53: • query-src-port-53—DNS packet source port 53	true | false
specify-query-stream-len	(Optional) Enables the query stream length: • query-stream-len—DNS Packet Length	0 to 65535
specify-query-type	(Optional) Enables the query type: • query-type—DNS Query Type 2 Byte Value	0 to 65535
specify-query-value	(Optional) Enables the query value: • query-value—Query 0 Response 1	true | false

Service FTP Engine

The Service FTP engine specializes in FTP port command decode, trapping invalid **port** commands and the PASV port spoof. It fills in the gaps when the String engine is not appropriate for detection. The parameters are Boolean and map to the various error trap conditions in the **port** command decode. The Service FTP engine runs on TCP ports 20 and 21. Port 20 is for data and the Service FTP engine does not do any inspection on this. It inspects the control transactions on port 21.

Table B-14 lists the parameters that are specific to the Service FTP engine.

Table B-14 Service FTP Engine Parameters

Parameter	Description	Value
direction	Direction of traffic: • Traffic from service port destined to client port • Traffic from client port destined to service port	from-service to-service
ftp-inspection-type	Type of inspection to perform: • Looks for an invalid address in the FTP port command • Looks for an invalid port in the FTP port command • Looks for the PASV port spoof	bad-port-cmd-address bad-port-cmd-port pasv
service-ports	A comma-separated list of ports or port ranges where the target service resides.	0 to 65535[1] a-b[,c-d]
swap-attacker-victim	Swaps the attacker and victim addresses and ports (source and destination) in the alert message and in any actions taken.	true | false (default)

1. The second number in the range must be greater than or equal to the first number.

Service Generic Engine

The Service Generic engine allows programmatic signatures to be issued in a config-file-only signature update. It has a simple machine and assembly language that is defined in the configuration file. It runs the machine code (distilled from the assembly language) through its virtual machine, which processes the instructions and pulls the important pieces of information out of the packet and runs them through the comparisons and operations specified in the machine code.

It is intended as a rapid signature response engine to supplement the String and State engines.

Note You cannot use the Service Generic engine to create custom signatures.

Caution Only advanced users should tune Service Generic engine signatures.

Table B-15 lists the parameters specific to the Service Generic engine.

Table B-15 Service Generic Engine Parameters

Parameter	Description	Value
specify-dst-port	(Optional) Enables the destination port: • dst-port—Destination port of interest for this signature	0 to 65535
specify-ip-protocol	(Optional) Enables IP protocol: • ip-protocol—The IP protocol this inspector should examine	0 to 255
specify-payload-source	(Optional) Enables payload source inspection: • payload-source—Payload source inspection for the following types: – Inspects ICMP data – Inspects Layer 2 headers – Inspects Layer 3 headers – Inspects Layer 4 headers – Inspects TCP data – Inspects UDP data	icmp-data l2-header l3-header l4-header tcp-data udp-data
specify-src-port	(Optional) Enables the source port: • src-port—Source port of interest for this signature	0 to 65535

Service H225 Engine

This section describes the Service H225 engine, and contains the following topics:

- Overview, page B-20
- Service H255 Engine Parameters, page B-22

Overview

The Service H225 engine analyzes H225.0 protocol, which consists of many subprotocols and is part of the H.323 suite. H.323 is a collection of protocols and other standards that together enable conferencing over packet-based networks.

H.225.0 call signaling and status messages are part of the H.323 call setup. Various H.323 entities in a network, such as the gatekeeper and endpoint terminals, run implementations of the H.225.0 protocol stack. The Service H225 engine analyzes H225.0 protocol for attacks on multiple H.323 gatekeepers, VoIP gateways, and endpoint terminals. It provides deep packet inspection for call signaling messages that are exchanged over TCP PDUs. The Service H225 engine analyzes the H.225.0 protocol for invalid H.255.0 messages, and misuse and overflow attacks on various protocol fields in these messages.

H.225.0 call signaling messages are based on Q.931 protocol. The calling endpoint sends a Q.931 setup message to the endpoint that it wants to call, the address of which it procures from the admissions procedure or some lookup means. The called endpoint either accepts the connection by transmitting a

Q.931 connect message or rejects the connection. When the H.225.0 connection is established, either the caller or the called endpoint provides an H.245 address, which is used to establish the control protocol (H.245) channel.

Especially important is the SETUP call signaling message because this is the first message exchanged between H.323 entities as part of the call setup. The SETUP message uses many of the commonly found fields in the call signaling messages, and implementations that are exposed to probable attacks will mostly also fail the security checks for the SETUP messages. Therefore, it is highly important to check the H.225.0 SETUP message for validity and enforce checks on the perimeter of the network.

The Service H225 engine has built-in signatures for TPKT validation, Q.931 protocol validation, and ASN.1PER validations for the H225 SETUP message. ASN.1 is a notation for describing data structures. PER uses a different style of encoding. It specializes the encoding based on the data type to generate much more compact representations.

You can tune the Q.931 and TPKT length signatures and you can add and apply granular signatures on specific H.225 protocol fields and apply multiple pattern search signatures of a single field in Q.931 or H.225 protocol.

The Service H225 engine supports the following features:

- TPKT validation and length check
- Q.931 information element validation
- Regular expression signatures on text fields in Q.931 information elements
- Length checking on Q.931 information elements
- SETUP message validation
- ASN.1 PER encode error checks
- Configuration signatures for fields like ULR-ID, E-mail-ID, h323-id, and so forth for both regular expression and length.

There is a fixed number of TPKT and ASN.1 signatures. You cannot create custom signatures for these types. For TPKT signatures, you should only change the value-range for length signatures. You should not change any parameters for ASN.1. For Q.931 signatures, you can add new regular expression signatures for text fields. for SETUP signatures, you can add signatures for length and regular expression checks on various SETUP message fields.

Service H255 Engine Parameters

Table B-16 lists parameters specific to the Service H225 engine.

Table B-16 Service H.225 Engine Parameters

Parameter	Description	Value
message-type	Type of H225 message to which the signature applies: • SETUP • ASN.1-PER • Q.931 • TPKT	asn.1-per q.931 setup tpkt
policy-type	Type of H225 policy to which the signature applies: • Inspects field length. • Inspects presence. If certain fields are present in the message, an alert is sent. • Inspects regular expressions. • Inspects field validations. • Inspects values. Regex and presence are not valid for TPKT signatures.	length presence regex validate value
specify-field-name	(Optional) Enables field name for use. Only valid for SETUP and Q.931 message types. Gives a dotted representation of the field name that this signature applies to. • field-name—Field name to inspect.	1 to 512
specify-invalid-packet-index	(Optional) Enables invalid packet index for use for specific errors in ASN, TPKT, and other errors that have fixed mapping. • invalid-packet-index—Inspection for invalid packet index.	0 to 255
specify-regex-string	The regular expression to look for when the policy type is regex. This is never set for TPKT signatures: • A regular expression to search for in a single TCP packet • (Optional) Enables min match length for use. The minimum length of the Regex match required to constitute a match. This is never set for TPKT signatures.	regex-string specify-min-match-length
specify-value-range	Valid for the length or value policy types (0x00 to 6535). Not valid for other policy types. • value-range—Range of values.	0 to 65535[1] a-b

1. The second number in the range must be greater than or equal to the first number.

Service HTTP Engine

This section describes the Service HTTP engine, and contains the following topics:

Overview

The Service HTTP engine is a service-specific string-based pattern-matching inspection engine. The HTTP protocol is one of the most commonly used in today's networks. In addition, it requires the most amount of preprocessing time and has the most number of signatures requiring inspection making it critical to the system's overall performance.

The Service HTTP engine uses a Regex library that can combine multiple patterns into a single pattern-matching table allowing a single search through the data. This engine searches traffic directed to web services only to web services, or HTTP requests. You cannot inspect return traffic with this engine. You can specify separate web ports of interest in each signature in this engine.

HTTP deobfuscation is the process of decoding an HTTP message by normalizing encoded characters to ASCII equivalent characters. It is also known as ASCII normalization.

Before an HTTP packet can be inspected, the data must be deobfuscated or normalized to the same representation that the target system sees when it processes the data. It is ideal to have a customized decoding technique for each host target type, which involves knowing what operating system and web server version is running on the target. The Service HTTP engine has default deobfuscation behavior for the Microsoft IIS web server.

Service HTTP Engine Parameters

For an example Service HTTP custom signature, see Example Service HTTP Signature, page 7-36.

Table B-17 lists the parameters specific the Service HTTP engine.

Table B-17 Service HTTP Engine Parameters

Parameter	Description	Value
de-obfuscate	Applies anti-evasive deobfuscation before searching.	true | false
max-field-sizes	Maximum field sizes grouping.	—
specify-max-arg-field-length	(Optional) Enables maximum argument field length: • max-arg-field-length—Maximum length of the arguments field.	0 to 65535
specify-max-header-field-length	(Optional) Enables maximum header field length: • max-header-field-length—Maximum length of the header field.	0 to 65535
specify-max-request-length	(Optional) Enables maximum request field length: • max-request-length—Maximum length of the request field.	0 to 65535

Table B-17 Service HTTP Engine Parameters (continued)

Parameter	Description	Value
specify-max-uri-field-length	(Optional) Enables the maximum URI field length: • max-uri-field-length—Maximum length of the URI field.	0 to 65535
regex	Regular expression grouping.	—
specify-arg-name-regex	(Optional) Enables searching the Arguments field for a specific regular expression: • arg-name-regex—Regular expression to search for in the HTTP Arguments field (after the ? and in the Entity body as defined by Content-Length).	—
specify-header-regex	(Optional) Enables searching the Header field for a specific regular expression: • header-regex—Regular Expression to search in the HTTP Header field. The Header is defined after the first CRLF and continues until CRLFCRLF.	—
specify-request-regex	(Optional) Enables searching the Request field for a specific regular expression: • request-regex—Regular expression to search in both HTTP URI and HTTP Argument fields. • specify-min-request-match-length—Enables setting a minimum request match length.	0 to 65535
specify-uri-regex	(Optional) Regular expression to search in HTTP URI field. The URI field is defined to be after the HTTP method (GET, for example) and before the first CRLF. The regular expression is protected, which means you cannot change the value.	[/\\][a-zA-Z][a-zA-Z][a-zA-Z][a-zA-Z][a-zA-Z][a-zA-Z][a-zA-Z][a-zA-Z][.]jpeg
service-ports	A comma-separated list of ports or port ranges where the target service resides.	0 to 65535[1] a-b[,c-d]
swap-attacker-victim	Swaps the attacker and victim addresses and ports (source and destination) in the alert message and in any actions taken.	true | false (default)

1. The second number in the range must be greater than or equal to the first number.

Service IDENT Engine

The Service IDENT engine inspects TCP port 113 traffic. It has basic decode and provides parameters to specify length overflows.

Table B-18 lists the parameters specific to the Service IDENT engine.

Table B-18 Service IDENT Engine Parameters

Parameter	Description	Value
inspection-type	Type of inspection to perform.	—
has-bad-port	Inspects payload for a bad port.	true | false
has-newline	Inspects payload for a nonterminating new line character.	true | false
size	Inspects for payload length longer than this.	0 to 65535
service-ports	A comma-separated list of ports or port ranges where the target service resides.	0 to 65535[1] a-b[,c-d]
direction	Direction of the traffic: • Traffic from service port destined to client port. • Traffic from client port destined to service port.	from-service to-service

1. The second number in the range must be greater than or equal to the first number.

Service MSRPC Engine

This section describes the Service MSRPC engine, and contains the following topics:

- Overview, page B-25
- Service MSRPC Engine Parameters, page B-26

Overview

The Service MSRPC engine processes MSRPC packets. MSRPC allows for cooperative processing between multiple computers and their application software in a networked environment. It is a transaction-based protocol, implying that there is a sequence of communications that establish the channel and pass processing requests and replies.

MSRPC is an ISO layer 5-6 protocol and is layered on top of other transport protocols such as UDP, TCP, and SMB. The MSRPC engine contains facilities to allow for fragmentation and reassembly of the MSRPC PDUs.

This communication channel is the source of recent Windows NT, Windows 2000, and Window XP security vulnerabilities.

The Service MSRPC engine only decodes the DCE and RPC protocol for the most common transaction types.

Service MSRPC Engine Parameters

Table B-19 lists the parameters specific to the Service MSRPC engine.

Table B-19 Service MSRPC Engine Parameters

Parameter	Description	Value
protocol	Protocol of interest for this inspector.	tcp udp
specify-operation	(Optional) Enables using MSRPC operation: • operation—MSRPC operation requested. Required for SMB_COM_TRANSACTION commands. Exact match.	0 to 65535
specify-regex-string	(Optional) Enables using a regular expression string: • specify-exact-match-offset—Enables the exact match offset: – exact-match-offset—The exact stream offset the regular expression string must report for a match to be valid. • specify-min-match-length—Enables the minimum match length: – min-match-length—Minimum number of bytes the regular expression string must match.	0 to 65535
specify-uuid	(Optional) Enables UUID: • uuid—MSRPC UUID field.	000001a0000 00000c00000 0000000046

Service MSSQL Engine

The Service MSSQL engine inspects the protocol used by Microsoft's SQL server (MSSQL).

There is one MSSQL signature. It fires an alert when it detects an attempt to log in to an MSSQL server with the default sa account.

You can add custom signatures based on MSSQL protocol values, such as login username and whether a password was used.

Table B-20 lists the parameters specific to the Service MSSQL engine.

Table B-20 Service MSSQL Engine Parameters

Parameter	Description	Value
password-present	Whether or not a password was used in an MS SQL login.	true | false
specify-sql-username	(Optional) Enables using an SQL username: • sql-username—Username (exact match) of user logging in to MS SQL service.	sa

Service NTP Engine

The Service NTP engine inspects NTP protocol. There is one NTP signature, the NTPd readvar overflow signature, which fires an alert if a readvar command is seen with NTP data that is too large for the NTP service to capture.

You can tune this signature and create custom signatures based on NTP protocol values, such as mode and size of control packets.

Table B-21 lists the parameters specific to the Service NTP engine.

Table B-21 Service NTP Engine Parameters

Parameter	Description	Value
inspection-type	Type of inspection to perform.	
inspect-ntp-packets	Inspects NTP packets: • control-opcode—Opcode number of an NTP control packet according to RFC1305, Appendix B. • max-control-data-size—Maximum allowed amount of data sent in a control packet. • mode —Mode of operation of the NTP packet per RFC 1305.	0 to 65535
is-invalid-data-packet	Looks for invalid NTP data packets. Checks the structure of the NTP data packet to make sure it is the correct size.	true | false
is-non-ntp-traffic	Checks for nonNTP packets on an NTP port.	true | false

Service RPC Engine

The Service RPC engine specializes in RPC protocol and has full decode as an anti-evasive strategy. It can handle fragmented messages (one message in several packets) and batch messages (several messages in a single packet).

The RPC portmapper operates on port 111. Regular RPC messages can be on any port greater than 550. RPC sweeps are like TCP port sweeps, except that they only count unique ports when a valid RPC message is sent. RPC also runs on UDP.

Table B-22 lists the parameters specific to the Service RPC engine.

Table B-22 Service RPC Engine Parameters

Parameter	Description	Value
direction	Direction of traffic: • Traffic from service port destined to client port. • Traffic from client port destined to service port.	from-service to-service
protocol	Protocol of interest.	tcp udp
service-ports	A comma-separated list of ports or port ranges where the target service resides.	0 to 65535[1] a-b[,c-d]

Table B-22 Service RPC Engine Parameters (continued)

Parameter	Description	Value
specify-is-spoof-src	(Optional) Enables the spoof source address: • is-spoof-src—Fires an alert when the source address is 127.0.0.1.	true | false
specify-port-map-program	(Optional) Enables the portmapper program: • port-map-program—The program number sent to the portmapper for this signature.	0 to 9999999999
specify-rpc-max-length	(Optional) Enables RPC maximum length: • rpc-max-length—Maximum allowed length of the entire RPC message. Lengths longer than what you specify fire an alert.	0 to 65535
specify-rpc-procedure	(Optional) Enables RPC procedure: • rpc-procedure—RPC procedure number for this signature.	0 to 1000000
specify-rpc-program	(Optional) Enables RPC program: • rpc-program—RPC program number for this signature.	0 to 1000000

1. The second number in the range must be greater than or equal to the first number.

Service SMB Engine

The Service SMB engine inspects SMB packets. You can tune SMB signatures and create custom SMB signatures based on SMB control transaction exchanges and SMB NT_Create_AndX exchanges.

Table B-23 lists the parameters specific to the Service SMB engine.

Table B-23 Service SMB Engine Parameters

Parameter	Description	Value
service-ports	A comma-separated list of ports or port ranges where the target service resides.	0 to 65535 a-b[,c-d] [1]
specify-allocation-hint	(Optional) Enables MSRPC allocation hint: • allocation-hint—MSRPC Allocation Hint, which is used in SMB_COM_TRANSACTION command parsing.[2]	0 to 42949677295
specify-byte-count	(Optional) Enables byte count: • byte-count—Byte count from SMB_COM_TRANSACTION structure.[3]	0 to 65535
specify-command	(Optional) Enables SMB commands: • command—SMB command value.[4]	0 to 255

Table B-23 Service SMB Engine Parameters (continued)

Parameter	Description	Value
specify-direction	(Optional) Enables traffic direction: • direction—Lets you specify the direction of traffic: – Traffic from service port destined to client port. – Traffic from client port destined to service port.	from service to service
specify-file-id	(Optional) Enables using a transaction file ID: • file-id—Transaction File ID.[5] **Note** This parameter may limit a signature to a specific exploit instance and its use should be carefully considered.	0 to 65535
specify-function	(Optional) Enables named pipe function: • function—Named Pipe function.[6]	0 to 65535
specify-hit-count	(Optional) Enables hit counting: • hit-count—The threshold number of occurrences in scan-interval to fire alerts.[7]	0 to 65535
specify-operation	(Optional) Enables MSRPC operation: • operation—MSRPC operation requested. Required for SMB_COM_TRANSACTION commands. An exact match is required.	0 to 65535
specify-resource	(Optional) Enables resource: • resource—Specifies that pipe or the SMB filename is used to qualify the alert. In ASCII format. An exact match is required.	*resource*
specify-scan-interval	(Optional) Enables scan interval: • scan-interval—The interval in seconds used to calculate alert rates.[8]	0 to 131071
specify-set-count	(Optional) Enables counting setup words: • set-count—Number of Setup words.[9]	0 to 255
specify-type	(Optional) Enables searching for the Type field of an MSRPC packet: • type —Type Field of MSRPC packet. 0 = Request; 2 = Response; 11 = Bind; 12 = Bind Ack	0 to 255

Table B-23 Service SMB Engine Parameters (continued)

Parameter	Description	Value
specify-word-count	(Optional) Enables word counting for command parameters: • word-count—Word count for the SMB_COM_TRANSACTION command parameters.[10]	0 to 255
swap-attacker-victim	Swaps the attacker and victim addresses and ports (source and destination) in the alert message and in any actions taken.	true | false (default)

1. The second number in the range must be greater than or equal to the first number.
2. An exact match is optional.
3. An exact match is optional.
4. An exact match is required. Currently supporting the 37 (0x25) SMB_COM_TRANSACTION command \x26amp and the 162 (0xA2) SMB_COM_NT_CREATE_ANDX command.
5. An exact match is optional.
6. An exact match is required. Required for SMB_COM_TRANSACTION commands.
7. Valid for signatures 3302 and 6255 only.
8. Valid for signatures 3302 and 6255 only.
9. An exact match is required. Usually two are required for SMB_COM_TRANSACTION commands.
10. An exact match is required. Only 16 word transactions are decoded.

Service SNMP Engine

The Service SNMP engine inspects all SNMP packets destined for port 161. You can tune SNMP signatures and create custom SNMP signatures based on specific community names and object identifiers.

Instead of using string comparison or regular expression operations to match the community name and object identifier, all comparisons are made using the integers to speed up the protocol decode and reduce storage requirements.

Table B-24 lists the parameters specific to the Service SNMP engine.

Table B-24 Service SNMP Engine Parameters

Parameter	Description	Value
inspection-type	Type of inspection to perform.	—
brute-force-inspection	Inspects for brute force attempts: • brute-force-count—The number of unique SNMP community names that constitute a brute force attempt.	0 to 65535
invalid-packet-inspection	Inspects for SNMP protocol violations.	—

Table B-24 Service SNMP Engine Parameters (continued)

Parameter	Description	Value
non-snmp-traffic-inspection	Inspects for non-SNMP traffic destined for UDP port 161.	—
snmp-inspection	Inspects SNMP traffic: • specify-community-name [yes | no]: – community-name—Searches for the SNMP community name, that is, the SNMP password. • specify-object-id [yes | no]: – object-id—Searches for the SNMP object identifier.	*community-name* *object-id*

Service SSH Engine

The Service SSH engine specializes in port 22 SSH traffic. Because all but the setup of an SSH session is encrypted, the engine only looks at the fields in the setup. There are two default signatures for SSH. You can tune these signatures, but you cannot create custom signatures.

Table B-25 lists the parameters specific to the Service SSH engine.

Table B-25 Service SSH Engine Parameters

Parameter	Description	Value
length-type	Inspects for one of the following SSH length types: • key-length—Length of the SSH key to inspect for: – length—Keys larger than this fire the RSAREF overflow. • user-length—User length SSH inspection: – length—Keys larger than this fire the RSAREF overflow.	0 to 65535
service-ports	A comma-separated list of ports or port ranges where the target service resides.	0 to 65535[1] a-b[,c-d]
specify-packet-depth	(Optional) Enables packet depth: • packet-depth—Number of packets to watch before determining the session key was missed.	0 to 65535

1. The second number in the range must be greater than or equal to the first number.

State Engine

The State engine provides state-based regular expression-based pattern inspection of TCP streams. A state engine is a device that stores the state of something and at a given time can operate on input to transition from one state to another and/or cause an action or output to take place. State machines are used to describe a specific event that causes an output or alert. There are three state machines in the State engine: SMTP, Cisco Login, and LPR Format String.

Table B-26 lists the parameters specific to the State engine.

Table B-26 State Engine Parameters

Parameter	Description	Value
state-machine	State machine grouping.	—
cisco-login	Specifies the state machine for Cisco login: • state-name—Name of the state required before the signature fires an alert: – Cisco device state – Control-C state – Password prompt state – Start state	cisco-device control-c pass-prompt start
lpr-format-string	Specifies the state machine to inspect for the LPR format string vulnerability: • state-name—Name of the state required before the signature fires an alert: – Abort state to end LPR Format String inspection – Format character state – State state	abort format-char start
smtp	Specifies the state machine for the SMTP protocol: • state-name—Name of the state required before the signature fires an alert: – Abort state to end LPR Format String inspection – Mail body state – Mail header state – SMTP commands state – Start state	abort mail-body mail-header smtp-commands start
direction	Direction of the traffic: • Traffic from service port destined to client port. • Traffic from client port destined to service port.	from-service to-service
service-ports	A comma-separated list of ports or port ranges where the target service resides.	0 to 65535 a-b[,c-d] [1]

Table B-26 State Engine Parameters (continued)

Parameter	Description	Value
specify-exact-match-offset	(Optional) Enables exact match offset: • exact-match-offset—The exact stream offset the regular expression string must report for a match to be valid.	0 to 65535
specify-min-match-length	(Optional) Enables minimum match length: • min-match-length—Minimum number of bytes the regular expression string must match.	0 to 65535
swap-attacker-victim	Swaps the attacker and victim addresses and ports (source and destination) in the alert message and in any actions taken.	true | false (default)

1. The second number in the range must be greater than or equal to the first number.

String Engines

This section describes the String engine, and contains the following topics:

Overview

The String engine is a generic-based pattern-matching inspection engine for ICMP, TCP, and UDP protocols. The String engine uses a regular expression engine that can combine multiple patterns into a single pattern-matching table allowing for a single search through the data. There are three String engines: String ICMP, String TCP, and String UDP.

String ICMP Engine Parameters

For an example custom String engine signature, see Example String TCP Signature, page 7-33.

Table B-27 lists the parameters specific to the String ICMP engine.

Table B-27 String ICMP Engine Parameters

Parameter	Description	Value
direction	Direction of the traffic: • Traffic from service port destined to client port. • Traffic from client port destined to service port.	from-service to-service
icmp-type	ICMP header TYPE value.	0 to 18[1] a-b[,c-d]

Table B-27 String ICMP Engine Parameters (continued)

Parameter	Description	Value
specify-exact-match-offset	(Optional) Enables exact match offset: • exact-match-offset—The exact stream offset the regular expression string must report for a match to be valid.	0 to 65535
specify-min-match-length	(Optional) Enables minimum match length: • min-match-length—Minimum number of bytes the regular expression string must match.	0 to 65535
swap-attacker-victim	Swaps the attacker and victim addresses and ports (source and destination) in the alert message and in any actions taken.	true | false (default)

1. The second number in the range must be greater than or equal to the first number.

String TCP Engine Parameters

For an example custom String engine signature, see Example String TCP Signature, page 7-33.

Table B-28 lists the parameters specific to the String TCP engine.

Table B-28 String TCP Engine

Parameter	Description	Value
direction	Direction of the traffic: • Traffic from service port destined to client port. • Traffic from client port destined to service port.	from-service to-service
service-ports	A comma-separated list of ports or port ranges where the target service resides.	0 to 65535[1] a-b[,c-d]
specify-exact-match-offset	(Optional) Enables exact match offset: • exact-match-offset—The exact stream offset the regular expression string must report for a match to be valid.	0 to 65535
specify-min-match-length	(Optional) Enables minimum match length: • min-match-length—Minimum number of bytes the regular expression string must match.	0 to 65535
strip-telnet-options	Strips the Telnet option characters from the data before the pattern is searched.[2]	true | false
swap-attacker-victim	Swaps the attacker and victim addresses and ports (source and destination) in the alert message and in any actions taken.	true | false (default)

1. The second number in the range must be greater than or equal to the first number.
2. This parameter is primarily used as an IPS anti-evasion tool.

String UDP Engine Parameters

For an example custom String engine signature, see Example String TCP Signature, page 7-33.

Table B-29 lists the parameters specific to the String UDP engine.

Table B-29 String UDP Engine

Parameter	Description	Value
direction	Direction of the traffic: • Traffic from service port destined to client port. • Traffic from client port destined to service port.	from-service to-service
service-ports	A comma-separated list of ports or port ranges where the target service resides.	0 to 65535[1] a-b[,c-d]
specify-exact-match-offset	(Optional) Enables exact match offset: • exact-match-offset—The exact stream offset the regular expression string must report for a match to be valid.	0 to 65535
specify-min-match-length	(Optional) Enables minimum match length: • min-match-length—Minimum number of bytes the regular expression string must match.	0 to 65535
swap-attacker-victim	Swaps the attacker and victim addresses and ports (source and destination) in the alert message and in any actions taken.	true | false (default)

1. The second number in the range must be greater than or equal to the first number.

Sweep Engine

The Sweep engine analyzes traffic between two hosts or from one host to many hosts. You can tune the existing signatures or create custom signatures. The Sweep engine has protocol-specific parameters for ICMP, UDP, and TCP.

The alert conditions of the Sweep engine ultimately depend on the count of the unique parameter. The unique parameter is the threshold number of distinct hosts or ports depending on the type of sweep. The unique parameter triggers the alert when more than the unique number of ports or hosts is seen on the address set within the time period. The processing of unique port and host tracking is called counting.

You can configure source and destination address filters, which means the sweep signature will exclude these addresses from the sweep-counting algorithm.

Caution Event action filters based on source and destination IP addresses do not function for the Sweep engine, because they do not filter as regular signatures. To filter source and destination IP addresses in sweep alerts, use the source and destination IP address filter parameters in the Sweep engine signatures.

A unique parameter must be specified for all signatures in the Sweep engine. A limit of 2 through 40 (inclusive) is enforced on the sweeps. 2 is the absolute minimum for a sweep, otherwise, it is not a sweep (of one host or port). 40 is a practical maximum that must be enforced so that the sweep does not consume excess memory. More realistic values for unique range between 5 and 15.

TCP sweeps must have a TCP flag and mask specified to determine which sweep inspector slot in which to count the distinct connections. The ICMP sweeps must have an ICMP type specified to discriminate among the various types of ICMP packets.

DataNode

When an activity related to Sweep engine signatures is seen, the IPS uses a DataNode to determine when it should stop monitoring for a particular host. The DataNode contains various persistent counters and variables needed for cross-packet reassembly of streams and for tracking the inspection state on a per-stream/per-source/per-destination basis The DataNode containing the sweep determines when the sweep should expire. The DataNode stops a sweep when the DataNode has not seen any traffic for *x* number of seconds (depending on the protocol).

There are several adaptive timeouts for the DataNodes. The DataNode expires after 30 seconds of idle time on the address set after all of the contained objects have been removed. Each contained object has various timeouts, for example, TCP Stream has a one-hour timeout for established connections. Most other objects have a much shorter expiration time, such as 5 or 60 seconds.

Table B-30 lists the parameters specific to the Sweep engine.

Table B-30 Sweep Engine Parameters

Parameter	Description	Value
dst-addr-filter	Destination IP address to exclude from the sweep counting algorithm.	<A.B.C.D>-<A.B.C.D> [,<A.B.C.D>-<A.B.C.D>]
src-addr-filter	Source IP address to exclude from the sweep counting algorithm.	<A.B.C.D>-<A.B.C.D> [,<A.B.C.D>-<A.B.C.D>]
protocol	Protocol of interest for this inspector.	icmp udp tcp
specify-icmp-type	(Optional) Enables inspection of the ICMP header type: • icmp-type—Specifies the ICMP header TYPE value.	0 to 255
specify-port-range	(Optional) Enables using a port range for inspection: • port-range—Specifies the UDP port range used in inspection.	0 to 65535 a-b[,c-d]
fragment-status	Specifies whether fragments are wanted or not: • Any fragment status. • Do not inspect fragments. • Inspect fragments.	any no-fragments want-fragments
inverted-sweep	Uses source port instead of destination port for unique counting.	true | false

Table B-30 Sweep Engine Parameters (continued)

Parameter	Description	Value
mask	Mask used in TCP flags comparison: • URG bit • ACK bit • PSH bit • RST bit • SYN bit • FIN bit	urg ack psh rst syn fin
storage-key	Type of address key used to store persistent data: • Attacker address • Attacker and victim addresses • Attacker address and victim port	Axxx AxBx Axxb
suppress-reverse	Does not fire when a sweep has fired in the reverse direction on this address set.	true | false
swap-attacker-victim	Swaps the attacker and victim addresses and ports (source and destination) in the alert message and in any actions taken.	true | false (default)
tcp-flags	TCP flags to match when masked by mask: • URG bit • ACK bit • PSH bit • RST bit • SYN bit • FIN bit	urg ack psh rst syn fin
unique	Threshold number of unique port connections between the two hosts.	0 to 65535

Traffic ICMP Engine

The Traffic ICMP engine analyzes nonstandard protocols, such as TFN2K, LOKI, and DDOS. There are only two signatures (based on the LOKI protocol) with user-configurable parameters.

Tribe Flood Net 2000 (TFN2K) is the newer version of the TFN. It is a Distributed Denial Of Service (DDoS) agent that is used to control coordinated attacks by infected computers (zombies) to target a single computer (or domain) with bogus traffic floods from hundreds or thousands of unknown attacking hosts. TFN2K sends randomized packet header information, but it has two discriminators that can be used to define signatures. One is whether the L3 checksum is incorrect and the other is whether the character 64 ‘A’ is found at the end of the payload. TFN2K can run on any port and can communicate with ICMP, TCP, UDP, or a combination of these protocols.

LOKI is a type of back door Trojan. When the computer is infected, the malicious code creates an "Icmp Tunnel" that can be used to send small payload in ICMP replies (which may go straight through a firewall if it is not configured to block ICMP.) The LOKI signatures look for an imbalance of ICMP echo requests to replies and simple ICMP code and payload discriminators.

The DDoS category (excluding TFN2K) targets ICMP-based DDoS agents. The main tools used here are TFN (Tribe Flood Net) and Stacheldraht. They are similar in operation to TFN2K, but rely on ICMP only and have fixed commands: integers and strings.

Table B-31 lists the parameters specific to the Traffic ICMP engine.

Table B-31 TRAFFIC.ICMP Engine Parameters

Parameter	Description	Value
parameter-tunable-sig	Whether this signature has configurable parameters.	yes | no
inspection-type	Type of inspection to perform: • Inspects for original LOKI traffic. • Inspects for modified LOKI traffic.	is-loki is-mod-loki
reply-ratio	Inbalance of replies to requests. The alert fires when there are this many more replies than requests.	0 to 65535
want-request	Requires an ECHO REQUEST be seen before firing the alert.	true | false

Trojan Engines

The Trojan engines analyze nonstandard protocols, such as BO2K andTFN2K. There are three Trojan engines: Trojan BO2K, Trojan TFN2K, and Trojan UDP.

BackOrifice (BO) was the original Windows back door Trojan that ran over UDP only. It was soon superseded by BackOrifice 2000 (BO2K). BO2K supported UDP and TCP both with basic XOR encryption. They have plain BO headers that have certain cross-packet characteristics.

BO2K also has a stealthy TCP module that was designed to encrypt the BO header and make the cross-packet patterns nearly unrecognizable. The UDP modes of BO and BO2K are handled by the Trojan UDP engine. The TCP modes are handled by the Trojan BO2K engine.

Note There are no specific parameters to the Trojan engines, except for swap-attacker-victim in the Trojan UDP engine.

APPENDIX C

Troubleshooting

Caution The BIOS on Cisco IDS/IPS sensors is specific to Cisco IDS/IPS sensors and must only be upgraded under instructions from Cisco with BIOS files obtained from the Cisco website. Installing a non-Cisco or third-party BIOS on Cisco IDS/IPS sensors voids the warranty. For more information on how to obtain instructions and BIOS files from the Cisco website, see Obtaining Obtaining Cisco IPS Software, page 18-1.

This appendix contains troubleshooting tips and procedures for sensors and software. It contains the following sections:

Bug Toolkit

For the most complete and up-to-date list of caveats, use the Bug Toolkit to refer to the caveat release note. You can use the Bug Toolkit to search for known bugs based on software version, feature set, and keywords. The resulting matrix shows when each bug was integrated, or fixed if applicable. It also lets you save the results of a search in Bug Groups, and also create persistent Alert Agents that can feed those groups with new defect alerts.

Note You must be logged in to Cisco.com to access the Bug Toolkit.

If you are a registered Cisco.com user, you can view the Bug Toolkit at this URL:

http://tools.cisco.com/Support/BugToolKit/action.do?hdnAction=searchBugs

To become a registered cisco.com user, go to this URL:

http://tools.cisco.com/RPF/register/register.do

Preventive Maintenance

The following actions will help you maintain your sensor:

- Back up a good configuration. If your current configuration becomes unusable, you can replace it with the backup version.

  For the procedure, see Creating and Using a Backup Configuration File, page 12-18.

- Save your backup configuration to a remote system.

  For the procedure, see Copying and Restoring the Configuration File Using a Remote Server, page 12-16.

- Always back up your configuration before you do a manual upgrade. If you have auto upgrades configured, make sure you do periodic backups.

- Create a service account.

  A service account is needed for password recovery and other special debug situations directed by TAC.

  For the procedure, see Creating the Service Account, page 4-13.

Caution You should carefully consider whether you want to create a service account. The service account provides shell access to the system, which makes the system vulnerable. However, you can use the service account to create a new password if the Administrator password is lost. Analyze your situation to decide if you want a service account existing on the system.

Note You cannot use the service account for password recovery on AIP-SSM, because you cannot get shell access to AIP-SSM. You must use ROMMON to get shell access to AIP-SSM.

Disaster Recovery

This section provides recommendations and steps to take if you need to recover your sensor after a disaster.

Follow these recommendations so that you are ready in case of a disaster:

- If you are using the CLI or IDM for configuration, copy the current configuration from the sensor to an FTP or SCP server any time a change has been made.

  For the procedure, see Creating and Using a Backup Configuration File, page 12-18.

  Note You should note the specific software version for that configuration. You can apply the copied configuration only to a sensor of the same version.

Note You also need the list of user IDs that have been used on that sensor. The list of user IDs and passwords are not saved in the configuration. For the procedure for obtaining a list of the current users on the sensor, see Viewing User Status, page 4-16.

- If you are using IDS MC, the current configuration is saved in the IDS MC database and a separate copy is not needed.

Note The list of user IDs is not saved in the IDS MC database. You must make a note of the user IDs.

Note You should note the specific software version for that configuration. You can push the copied configuration only to a sensor of the same version.

When a disaster happens and you need to recover the sensor, try the following:

1. Reimage the sensor.

   For the procedures for appliances and modules, see Chapter 17, "Upgrading, Downgrading, and Installing System Images."

2. Log in to the sensor with the default user ID and password—**cisco**.

   Note You are prompted to change the cisco password.

3. Run the **setup** command.

   For the procedure, see Initializing the Sensor, page 3-2.

4. Upgrade the sensor to the IPS software version it had when the configuration was last saved and copied.

   For more information on obtaining IPS software versions and how to install them, see Obtaining Cisco IPS Software, page 18-1.

Warning **Trying to copy the saved configuration without getting the sensor back to the same IPS software version it had before the disaster can cause configuration errors.**

5. Copy the last saved configuration to the sensor.

   For the procedure, see Creating and Using a Backup Configuration File, page 12-18.

6. Update clients to use the new key and certificate of the sensor.

   Reimaging changes the sensor's SSH keys and HTTPS certificate. For the procedure, see Adding Hosts to the Known Hosts List, page 4-32.

7. Create previous users.

   For the procedure, see Configuring User Parameters, page 4-11.

Password Recovery

The following password recovery options exist:

- If another Administrator account exists, the other Administrator can change the password.
- If a Service account exists, you can log in to the service account and switch to user root using the command **su - root**. Use the **password** command to change the CLI Administrator account's password. For example, if the Administrator username is "adminu," the command is **password adminu**. You are prompted to enter the new password twice. For more information, see Creating the Service Account, page 4-13.

You can reimage the sensor using either the recovery partition or a system image file. For more information, see Chapter 17, "Upgrading, Downgrading, and Installing System Images."

PIX 7.1 Devices and Normalizer Inline Mode

For IPS 5.0 and 5.1, normalizer inline mode may deny packets and/or connections if a PIX 7.1 device is in the traffic flow and the PIX device has been configured for the MSS workaround.

Certain web applications on port 80 cause the PIX device to require the MSS workaround. If that workaround is active, the IPS must have a complimentary workaround.

Problem There is an incompatibility with PIX and IPS when the PIX MSS workaround has been applied. The **show stat vi** command shows many deny packet or deny connection actions along with many 13*xx* signature firings.

Solution Disable or remove all actions from the following normalizer signatures: 1306 and 1311.

Time and the Sensor

This section describes how to maintain accurate time on the sensor, and contains the following topics:

Time Sources and the Sensor

The sensor requires a reliable time source. All events (alerts) must have the correct UTC and local time stamp, otherwise, you cannot correctly analyze the logs after an attack. When you initialize the sensor, you set up the time zones and summertime settings. For more information, see Initializing the Sensor, page 3-2.

Here is a summary of ways to set the time on sensors:

- For appliances
  - Use the **clock set** command to set the time. This is the default.

    For the procedure, see Manually Setting the Clock, page 13-8.

- Use NTP

  You can configure the appliance to get its time from an NTP time synchronization source. For more information, see Configuring a Cisco Router to be an NTP Server, page 4-29. You need the NTP server IP address, the NTP key ID, and the NTP key value. You can set up NTP on the appliance during initialization or you can configure NTP through the CLI, IDM, or ASDM.

  Note We recommend that you use an NTP time synchronization source.

- For IDSM-2
  - The IDSM-2 can automatically synchronize its clock with the switch time. This is the default.

    Note The UTC time is synchronized between the switch and the IDSM-2. The time zone and summertime settings are not synchronized between the switch and the IDSM-2.

Caution Be sure to set the time zone and summertime settings on both the switch and IDSM-2 to ensure that the UTC time settings are correct. The local time of IDSM-2 could be incorrect if the time zone and/or summertime settings do not match between IDSM-2 and the switch. For more information, see Synchronizing IPS Module System Clocks with Parent Device System Clocks, page C-6.

  - Use NTP

    You can configure IDSM-2 to get its time from an NTP time synchronization source. For more information, see Configuring a Cisco Router to be an NTP Server, page 4-29. You need the NTP server IP address, the NTP key ID, and the NTP key value. You can configure IDSM-2 to use NTP during initialization or you can set up NTP through the CLI, IDM, or ASDM.

    Note We recommend that you use an NTP time synchronization source.

- For NM-CIDS
  - NM-CIDS can automatically synchronize its clock with the clock in the router chassis in which it is installed (parent router). This is the default.

    Note The UTC time is synchronized between the parent router and NM-CIDS. The time zone and summertime settings are not synchronized between the parent router and NM-CIDS.

Caution Be sure to set the time zone and summertime settings on both the parent router and NM-CIDS to ensure that the UTC time settings are correct. The local time of NM-CIDS could be incorrect if the time zone and/or summertime settings do not match between NM-CIDS and the router. For more information, see Synchronizing IPS Module System Clocks with Parent Device System Clocks, page C-6.

– Use NTP

You can configure NM-CIDS to get its time from an NTP time synchronization source, such as a Cisco router other than the parent router. For more information, see Configuring a Cisco Router to be an NTP Server, page 4-29. You need the NTP server IP address, the NTP key ID, and the NTP key value. You can configure NM-CIDS to use NTP during initialization or you can set up NTP through the CLI, IDM, or ASDM.

Note We recommend that you use an NTP time synchronization source.

- For AIP-SSM
  – AIP-SSM can automatically synchronize its clock with the clock in the adaptive security appliance in which it is installed. This is the default.

Note The UTC time is synchronized between the adaptive security appliance and AIP-SSM. The time zone and summertime settings are not synchronized between the adaptive security appliance and AIP-SSM.

Caution Be sure to set the time zone and summertime settings on both the adaptive security appliance and AIP-SSM to ensure that the UTC time settings are correct. The local time of AIP-SSM could be incorrect if the time zone and/or summertime settings do not match between AIP-SSM and the adaptive security appliance. For more information, see Synchronizing IPS Module System Clocks with Parent Device System Clocks, page C-6.

– Use NTP

You can configure AIP-SSM to get its time from an NTP time synchronization source, such as a Cisco router other than the parent router. For more information, see Configuring a Cisco Router to be an NTP Server, page 4-29. You need the NTP server IP address, the NTP key ID, and the NTP key value. You can configure AIP-SSM to use NTP during initialization or you can set up NTP through the CLI, IDM, or ASDM.

Note We recommend that you use an NTP time synchronization source.

Synchronizing IPS Module System Clocks with Parent Device System Clocks

All IPS modules (IDSM-2, NM-CIDS, and AIP-SSM) synchronize their system clocks to the parent chassis clock (switch, router, or firewall) each time the module boots up and any time the parent chassis clock is set. The module clock and parent chassis clock tend to drift apart over time. The difference can be as much as several seconds per day. To avoid this problem, make sure that both the module clock and the parent clock are synchronized to an external NTP server. If only the module clock or only the parent chassis clock is synchronized to an NTP server, the time drift occurs. For more information on NTP, see Configuring NTP, page 4-29. For more information on verifying that the module and NTP server are synchronized, see Verifying the Sensor is Synchronized with the NTP Server, page C-7.

Verifying the Sensor is Synchronized with the NTP Server

In IPS 5.1, you cannot apply an incorrect NTP configuration, such as an invalid NTP key value or ID, to the sensor. If you try to apply an incorrect configuration, you receive an error message. To verify the NTP configuration, use the **show statistics host** command to gather sensor statistics. The NTP statistics section provides NTP statistics including feedback on sensor synchronization with the NTP server.

To verify the NTP configuration, follow these steps:

Step 1 Log in to the sensor.

Step 2 Generate the host statistics:

```
sensor# show statistics host
     ...
     NTP Statistics
             remote           refid      st t when poll reach   delay   offset  jitter
         11.22.33.44     CHU_AUDIO(1)     8 u   36   64    1    0.536    0.069   0.001
         LOCAL(0)        73.78.73.84      5 l   35   64    1    0.000    0.000   0.001
        ind assID status  conf reach auth condition  last_event cnt
          1 10372  f014   yes   yes   ok     reject   reachable  1
          2 10373  9014   yes   yes  none    reject   reachable  1
        status = Not Synchronized
     ...
```

Step 3 Generate the hosts statistics again after a few minutes:

```
sensor# show statistics host
     ...
     NTP Statistics
             remote           refid      st t when poll reach   delay   offset  jitter
        *11.22.33.44     CHU_AUDIO(1)     8 u   22   64  377    0.518   37.975  33.465
         LOCAL(0)        73.78.73.84      5 l   22   64  377    0.000    0.000   0.001
        ind assID status  conf reach auth condition  last_event cnt
          1 10372  f624   yes   yes   ok   sys.peer   reachable  2
          2 10373  9024   yes   yes  none    reject   reachable  2
        status = Synchronized
```

Step 4 If the status continues to read `Not Synchronized`, check with the NTP server administrator to make sure the NTP server is configured correctly.

Correcting Time on the Sensor

If you set the time incorrectly, your stored events will have the incorrect time because they are stamped with the time the event was created.

The Event Store time stamp is always based on UTC time. If during the original sensor setup, you set the time incorrectly by specifying 8:00 p.m. rather than 8:00 a.m., when you do correct the error, the corrected time will be set backwards. New events might have times older than old events.

For example, if during the initial setup, you configure the sensor as central time with daylight saving time enabled and the local time is 8:04 p.m., the time is displayed as 20:04:37 CDT and has an offset from UTC of -5 hours (01:04:37 UTC, the next day). A week later at 9:00 a.m., you discover the error: the clock shows 21:00:23 CDT. You then change the time to 9:00 a.m. and now the clock shows 09:01:33 CDT. Because the offset from UTC has not changed, it requires that the UTC time now be 14:01:33 UTC, which creates the time stamp problem.

To ensure the integrity of the time stamp on the event records, you must clear the event archive of the older events by using the **clear events** command. For more information on the **clear events** command, see Clearing Events from the Event Store, page 13-7.

Caution You cannot remove individual events.

Troubleshooting the 4200 Series Appliance

This section contains information to troubleshoot the 4200 series appliance.

Tip Before troubleshooting the appliance, check the Caveats section of the Readme for the software version you have installed on your sensor to see if you are dealing with a known issue.

This section contains the following topics:

Communication Problems

This section helps you troubleshoot communication problems with the 4200 series sensor. It contains the following topics:

Cannot Access the Sensor CLI Through Telnet or SSH

If you cannot access the sensor CLI through Telnet (if you already have it enabled) or SSH, follow these steps:

Note For the procedure for enabling and disabling Telnet on the sensor, see Enabling and Disabling Telnet, page 4-4.

Step 1 Log in to the sensor CLI through a console, terminal, or module session.

For the various ways to open a CLI session directly on the sensor, see Chapter 2, "Logging In to the Sensor."

Step 2 Make sure that the sensor management interface is enabled:

```
sensor# show interfaces
Interface Statistics
   Total Packets Received = 0
   Total Bytes Received = 0
   Missed Packet Percentage = 0
   Current Bypass Mode = Auto_off
MAC statistics from interface GigabitEthernet0/1
   Media Type = backplane
   Missed Packet Percentage = 0
   Inline Mode = Unpaired
   Pair Status = N/A
   Link Status = Up
   Link Speed = Auto_1000
   Link Duplex = Auto_Full
   Total Packets Received = 0
   Total Bytes Received = 0
   Total Multicast Packets Received = 0
   Total Broadcast Packets Received = 0
   Total Jumbo Packets Received = 0
   Total Undersize Packets Received = 0
   Total Receive Errors = 0
   Total Receive FIFO Overruns = 0
   Total Packets Transmitted = 0
   Total Bytes Transmitted = 0
   Total Multicast Packets Transmitted = 0
   Total Broadcast Packets Transmitted = 0
   Total Jumbo Packets Transmitted = 0
   Total Undersize Packets Transmitted = 0
   Total Transmit Errors = 0
   Total Transmit FIFO Overruns = 0
MAC statistics from interface GigabitEthernet0/0
   Media Type = TX
   Link Status = Up
   Link Speed = Auto_100
   Link Duplex = Auto_Full
   Total Packets Received = 944333
   Total Bytes Received = 83118358
   Total Multicast Packets Received = 0
   Total Receive Errors = 0
   Total Receive FIFO Overruns = 0
   Total Packets Transmitted = 397633
   Total Bytes Transmitted = 435730956
   Total Transmit Errors = 0
   Total Transmit FIFO Overruns = 0
sensor#
```

The management interface is the interface in the list with the status line `Media Type = TX`. If the Link Status is `Down`, go to Step 3. If the Link Status is `Up`, go to Step 5.

Step 3 Make sure the sensor's IP address is unique.

```
sensor# setup
--- System Configuration Dialog ---

At any point you may enter a question mark '?' for help.
User ctrl-c to abort configuration dialog at any prompt.
Default settings are in square brackets '[]'.


Current Configuration:
```

```
service host
network-settings
host-ip 10.89.130.108/23,10.89.130.1
host-name sensor
telnet-option enabled
access-list 0.0.0.0/0
ftp-timeout 300
no login-banner-text
exit
--MORE--
```

If the management interface detects that another device on the network has the same IP address, it will not come up.

For more information, see Changing the IP Address, Netmask, and Gateway, page 4-3.

Step 4 Make sure the management port is connected to an active network connection.

If the management port is not connected to an active network connection, the management interface will not come up.

Step 5 Make sure the IP address of the workstation that is trying to connect to the sensor is permitted in the sensor's access list:

```
sensor# setup
--- System Configuration Dialog ---

At any point you may enter a question mark '?' for help.
User ctrl-c to abort configuration dialog at any prompt.
Default settings are in square brackets '[]'.


Current Configuration:


service host
network-settings
host-ip 10.89.130.108/23,10.89.130.1
host-name sensor
telnet-option enabled
access-list 0.0.0.0/0
ftp-timeout 300
no login-banner-text
exit
--MORE--
```

If the network address of the workstation is permitted in the sensor access list, go to Step 6.

Step 6 Add a permit entry for the workstation's network address, save the configuration, and try to connect again.

For more information, see Changing the Access List, page 4-5.

Step 7 Make sure the network configuration allows the workstation to connect to the sensor.

If the sensor is protected behind a firewall and the workstation is in front of the firewall, make sure the firewall is configured to allow the workstation to access the sensor. Or if the workstation is behind a firewall that is performing network address translation on the IP address of the workstation, and the sensor is in front of the firewall, make sure that the sensor's access list contains a permit entry for the workstation's translated address.

For more information, see Changing the Access List, page 4-5.

Misconfigured Access List

To correct a misconfigured access list, follow these steps:

Step 1 Log in to the CLI.

Step 2 View your configuration to see the access list:

```
sensor# show configuration | include access-list
access-list 10.0.0.0/8
access-list 64.0.0.0/8
sensor#
```

Step 3 Verify that the client IP address is listed in the allowed networks. If it is not, add it:

```
sensor# configure terminal
sensor(config)# service host
sensor(config-hos)# network-settings
sensor(config-hos-net)# access-list 171.69.70.0/24
```

Step 4 Verify the settings:

```
sensor(config-hos-net)# show settings
   network-settings
   ----------------------------------------------
      host-ip: 10.89.149.238/25,10.89.149.254 default: 10.1.9.201/24,10.1.9.1
      host-name: qsensor-238 default: sensor
      telnet-option: enabled default: disabled
      access-list (min: 0, max: 512, current: 3)
      ---------------------------------------------
         network-address: 10.0.0.0/8
         ---------------------------------------------
         network-address: 64.0.0.0/8
         ---------------------------------------------
         network-address: 171.69.70.0/24
         ---------------------------------------------
      ---------------------------------------------
      ftp-timeout: 300 seconds <defaulted>
      login-banner-text: <defaulted>
   ----------------------------------------------
sensor(config-hos-net)#
```

Duplicate IP Address Shuts Interface Down

If you have two newly imaged sensors with the same IP address that come up on the same network at the same time, the interface shuts down. Linux prevents the command and control interface from activating if it detects an address conflict with another host.

To verify that the sensor in question does not have an IP address conflict with another host on the network, follow these steps:

Step 1 Log in to the CLI.

Step 2 Determine whether the interface is up:

```
sensor# show interfaces
Interface Statistics
   Total Packets Received = 0
   Total Bytes Received = 0
```

```
   Missed Packet Percentage = 0
   Current Bypass Mode = Auto_off
MAC statistics from interface GigabitEthernet0/1
   Media Type = backplane
   Missed Packet Percentage = 0
   Inline Mode = Unpaired
   Pair Status = N/A
   Link Status = Up
   Link Speed = Auto_1000
   Link Duplex = Auto_Full
   Total Packets Received = 0
   Total Bytes Received = 0
   Total Multicast Packets Received = 0
   Total Broadcast Packets Received = 0
   Total Jumbo Packets Received = 0
   Total Undersize Packets Received = 0
   Total Receive Errors = 0
   Total Receive FIFO Overruns = 0
   Total Packets Transmitted = 0
   Total Bytes Transmitted = 0
   Total Multicast Packets Transmitted = 0
   Total Broadcast Packets Transmitted = 0
   Total Jumbo Packets Transmitted = 0
   Total Undersize Packets Transmitted = 0
   Total Transmit Errors = 0
   Total Transmit FIFO Overruns = 0
MAC statistics from interface GigabitEthernet0/0
   Media Type = TX
   Link Status = Up
   Link Speed = Auto_100
   Link Duplex = Auto_Full
   Total Packets Received = 1822323
   Total Bytes Received = 131098876
   Total Multicast Packets Received = 20
   Total Receive Errors = 0
   Total Receive FIFO Overruns = 0
   Total Packets Transmitted = 219260
   Total Bytes Transmitted = 103668610
   Total Transmit Errors = 0
   Total Transmit FIFO Overruns = 0
sensor#
```

If the output says the command and control interface link status is down, there is a hardware issue or an IP address conflict.

Step 3 Make sure the cabling of the sensor is correct.

Refer to the chapter for your sensor in *Installing Cisco Intrusion Prevention System Appliances and Modules 5.1*.

Step 4 Run the **setup** command to make sure the IP address is correct.

For the procedure, see Initializing the Sensor, page 3-2.

SensorApp and Alerting

This section helps you troubleshoot issues with SensorApp and alerting. It contains the following topics:

SensorApp Not Running

The sensing process, SensorApp, should always be running. If it is not, you do not receive any alerts. SensorApp is part of Analysis Engine, so you must make sure the Analysis Engine is running.

To make sure Analysis Engine is running, follow these steps:

Step 1 Log in to the CLI.

Step 2 Determine the status of the Analysis Engine service:

```
sensor# show version
Application Partition:

Cisco Intrusion Prevention System, Version 5.0(1)S149.0

OS Version 2.4.26-IDS-smp-bigphys
Platform: ASA-SSM-20
Serial Number: 021
No license present
Sensor up-time is 19 days.
Using 505495552 out of 1984704512 bytes of available memory (25% usage)
system is using 17.3M out of 29.0M bytes of available disk space (59% usage)
application-data is using 37.7M out of 166.6M bytes of available disk space (24% usage)
boot is using 40.5M out of 68.5M bytes of available disk space (62% usage)


MainApp          2005_Mar_04_14.23   (Release)   2005-03-04T14:35:11-0600   Running
AnalysisEngine   2005_Mar_04_14.23   (Release)   2005-03-04T14:35:11-0600   Not Running
CLI              2005_Mar_04_14.23   (Release)   2005-03-04T14:35:11-0600


Upgrade History:

  IDS-K9-maj-5.0-1-   14:16:00 UTC Thu Mar 04 2004

Recovery Partition Version 1.1 - 5.0(1)S149

sensor#
```

Step 3 If Analysis Engine is not running, look for any errors connected to it:

```
sensor# show events error fatal past 13:00:00 | include AnalysisEngine
evError: eventId=1077219258696330005 severity=warning

originator:
hostId: sensor
appName: sensorApp
appInstanceId: 1045
time: 2004/02/19 19:34:20 2004/02/19 19:34:20 UTC
errorMessage: name=errUnclassified Generating new Analysis Engine configuration file.
```

Note The date and time of the last restart is listed. In this example, the last restart was on 2-19-2004 at 7:34.

Step 4 Make sure you have the latest software updates:

```
sensor# show version
Upgrade History:

  IDS-K9-maj-5.0-1-   14:16:00 UTC Thu Mar 04 2004

Recovery Partition Version 1.1 - 5.0(1)S149
```

If you do not have the latest software updates, download them from Cisco.com. For the procedure, see Obtaining Cisco IPS Software, page 18-1.

Step 5 Read the Readme that accompanies the software upgrade for any known DDTS for SensorApp or Analysis Engine.

Physical Connectivity, SPAN, or VACL Port Issue

If the sensor is not connected properly, you do not receive any alerts.

To make sure the sensor is connected properly, follow these steps:

Step 1 Log in to the CLI.

Step 2 Make sure the interfaces are up and that the packet count is increasing:

```
sensor# show interfaces
Interface Statistics
   Total Packets Received = 0
   Total Bytes Received = 0
   Missed Packet Percentage = 0
   Current Bypass Mode = Auto_off
MAC statistics from interface GigabitEthernet0/1
   Media Type = backplane
   Missed Packet Percentage = 0
   Inline Mode = Unpaired
   Pair Status = N/A
   Link Status = Up
   Link Speed = Auto_1000
   Link Duplex = Auto_Full
   Total Packets Received = 0
   Total Bytes Received = 0
   Total Multicast Packets Received = 0
   Total Broadcast Packets Received = 0
   Total Jumbo Packets Received = 0
   Total Undersize Packets Received = 0
   Total Receive Errors = 0
   Total Receive FIFO Overruns = 0
   Total Packets Transmitted = 0
   Total Bytes Transmitted = 0
   Total Multicast Packets Transmitted = 0
   Total Broadcast Packets Transmitted = 0
   Total Jumbo Packets Transmitted = 0
   Total Undersize Packets Transmitted = 0
   Total Transmit Errors = 0
```

```
   Total Transmit FIFO Overruns = 0
MAC statistics from interface GigabitEthernet0/0
   Media Type = TX
   Link Status = Up
   Link Speed = Auto_100
   Link Duplex = Auto_Full
   Total Packets Received = 1830137
   Total Bytes Received = 131624465
   Total Multicast Packets Received = 20
   Total Receive Errors = 0
   Total Receive FIFO Overruns = 0
   Total Packets Transmitted = 220052
   Total Bytes Transmitted = 103796666
   Total Transmit Errors = 0
   Total Transmit FIFO Overruns = 0
sensor#
```

Step 3 If the Link Status is down, make sure the sensing port is connected properly:

a. Make sure the sensing port is connected properly on the appliance.

Refer to the chapter on your appliance in *Installing Cisco Intrusion Prevention System Appliances and Modules 5.1*.

b. Make sure the sensing port is connected to the correct SPAN or VACL capture port on IDSM-2.

For more information, see Chapter 15, "Configuring IDSM-2."

Step 4 Verify the interface configuration:

a. Make sure you have the interfaces configured properly.

For the procedure, see Chapter 5, "Configuring Interfaces."

b. Verify the SPAN and VACL capture port configuration on the Cisco switch.

Refer to your switch documentation for the procedure.

Step 5 Verify again that the interfaces are up and that the packet count is increasing.

```
sensor# show interfaces
```

Unable to See Alerts

If you are not seeing alerts, try the following:

- Make sure the signature is enabled.
- Make sure the signature is not retired.
- Make sure that you have Produce Alert configured as an action.

Note If you choose Produce Alert, but come back later and add another event action and do not add Produce Alert to the new configuration, alerts are not be sent to the Event Store. Every time you configure a signature, the new configuration overwrites the old one, so make sure you have configured all the event actions you want for each signature.

- Make sure the sensor is seeing packets.
- Make sure that alerts are being generated.

To make sure you can see alerts, follow these steps:

Step 1 Log in to the CLI.

Step 2 Make sure the signature is enabled:

```
sensor# configure terminal
sensor(config)# service signature-definition sig0
sensor(config-sig)# signatures 1300 0
sensor(config-sig-sig)# status
sensor(config-sig-sig-sta)# show settings
   status
   ----------------------------------------------
      enabled: true <defaulted>
      retired: false <defaulted>
   ----------------------------------------------
sensor(config-sig-sig-sta)#
```

Step 3 Make sure you have Produce Alert configured:

```
sensor# configure terminal
sensor(config)# service signature-definition sig0
sensor(config-sig)# signatures 1300 0
sensor(config-sig-sig)# engine ?
normalizer     Signature engine
sensor(config-sig-sig)# engine normalizer
sensor(config-sig-sig-nor)# event-action produce-alert
sensor(config-sig-sig-nor)# show settings
   normalizer
   ----------------------------------------------
      event-action: produce-alert default: produce-alert|deny-connection-inline
      edit-default-sigs-only
      ----------------------------------------------
sensor#
```

Step 4 Make sure the sensor is seeing packets:

```
sensor# show interfaces FastEthernet0/1
MAC statistics from interface FastEthernet0/1
   Media Type = backplane
   Missed Packet Percentage = 0
   Inline Mode = Unpaired
   Pair Status = N/A
   Link Status = Up
   Link Speed = Auto_100
   Link Duplex = Auto_Full
   Total Packets Received = 267581
   Total Bytes Received = 24886471
   Total Multicast Packets Received = 0
   Total Broadcast Packets Received = 0
   Total Jumbo Packets Received = 0
   Total Undersize Packets Received = 0
   Total Receive Errors = 0
   Total Receive FIFO Overruns = 0
   Total Packets Transmitted = 57301
   Total Bytes Transmitted = 3441000
   Total Multicast Packets Transmitted = 0
   Total Broadcast Packets Transmitted = 0
   Total Jumbo Packets Transmitted = 0
   Total Undersize Packets Transmitted = 0
   Total Transmit Errors = 1
   Total Transmit FIFO Overruns = 0
sensor#
```

Step 5 Check for alerts:

```
sensor# show statistics virtual-sensor
SigEvent Preliminary Stage Statistics
    Number of Alerts received = 0
    Number of Alerts Consumed by AlertInterval = 0
    Number of Alerts Consumed by Event Count = 0
    Number of FireOnce First Alerts = 0
    Number of FireOnce Intermediate Alerts = 0
    Number of Summary First Alerts = 0
    Number of Summary Intermediate Alerts = 0
    Number of Regular Summary Final Alerts = 0
    Number of Global Summary Final Alerts = 0
    Number of Alerts Output for further processing = 0alertDetails: Traffic Source: int0;
```

Sensor Not Seeing Packets

If the sensor is not seeing any packets on the network, you could have the interfaces set up incorrectly.

If the sensor is not seeing packets, follow these steps:

Step 1 Log in to the CLI.

Step 2 Make sure the interfaces are up and receiving packets:

```
sensor# show interfaces GigabitEthernet0/1
MAC statistics from interface GigabitEthernet0/1
   Media Type = backplane
   Missed Packet Percentage = 0
   Inline Mode = Unpaired
   Pair Status = N/A
   Link Status = Down
   Link Speed = Auto_1000
   Link Duplex = Auto_Full
   Total Packets Received = 0
   Total Bytes Received = 0
   Total Multicast Packets Received = 0
   Total Broadcast Packets Received = 0
   Total Jumbo Packets Received = 0
   Total Undersize Packets Received = 0
   Total Receive Errors = 0
   Total Receive FIFO Overruns = 0
   Total Packets Transmitted = 0
   Total Bytes Transmitted = 0
   Total Multicast Packets Transmitted = 0
   Total Broadcast Packets Transmitted = 0
   Total Jumbo Packets Transmitted = 0
   Total Undersize Packets Transmitted = 0
   Total Transmit Errors = 0
   Total Transmit FIFO Overruns = 0
sensor#
```

Step 3 If the interfaces are not up, do the following:

a. Check the cabling.

For information on installing the sensor properly, refer to the chapter that pertains to your sensor in *Installing Cisco Intrusion Prevention System Appliances and Modules 5.1.*

b. Enable the interface.

```
sensor# configure terminal
sensor(config)# service interface
sensor(config-int)# physical-interfaces GigabitEthernet0/1
sensor(config-int-phy)# admin-state enabled
sensor(config-int-phy)# show settings
   <protected entry>
   name: GigabitEthernet0/1
   ----------------------------------------------
      media-type: tx <protected>
      description: <defaulted>
      admin-state: enabled default: disabled
      duplex: auto <defaulted>
      speed: auto <defaulted>
      alt-tcp-reset-interface
      ------------------------------------------------
         none
         ------------------------------------------------
         ------------------------------------------------
      ------------------------------------------------
   ----------------------------------------------
sensor(config-int-phy)#
```

Step 4 Check to see that the interface is up and receiving packets:

```
sensor# show interfaces
MAC statistics from interface GigabitEthernet0/1
   Media Type = TX
   Missed Packet Percentage = 0
   Inline Mode = Unpaired
   Pair Status = N/A
   Link Status = Up
   Link Speed = Auto_100
   Link Duplex = Auto_Full
   Total Packets Received = 3
   Total Bytes Received = 900
   Total Multicast Packets Received = 3
   Total Broadcast Packets Received = 0
   Total Jumbo Packets Received = 0
   Total Undersize Packets Received = 0
   Total Receive Errors = 0
   Total Receive FIFO Overruns = 0
   Total Packets Transmitted = 0
   Total Bytes Transmitted = 0
   Total Multicast Packets Transmitted = 0
   Total Broadcast Packets Transmitted = 0
   Total Jumbo Packets Transmitted = 0
   Total Undersize Packets Transmitted = 0
   Total Transmit Errors = 0
   Total Transmit FIFO Overruns = 0 ...
```

Cleaning Up a Corrupted SensorApp Configuration

If the SensorApp configuration has become corrupted and SensorApp cannot run, you must delete it entirely and restart SensorApp.

To delete the SensorApp configuration, follow these steps:

Step 1 Log in to the service account.

Step 2 Su to root.

Step 3 Stop the IPS applications:

```
/etc/init.d/cids stop
```

Step 4 Replace the virtual sensor file:

```
cp /usr/cids/idsRoot/etc/defVirtualSensorConfig.xml
/usr/cids/idsRoot/etc/VS-Config/virtualSensor.xml
```

Step 5 Remove the cache files:

```
rm /usr/cids/idsRoot/var/virtualSensor/*.pmz
```

Step 6 Exit the service account.

Step 7 Log in to the sensor CLI.

Step 8 Start the IPS services:

```
sensor# cids start
```

Step 9 Log in to an account with administrator privileges.

Step 10 Reboot the sensor:

```
sensor# reset
Warning: Executing this command will stop all applications and reboot the node.
Continue with reset? [yes]:yes
Request Succeeded.
sensor#
```

Bad Memory on IDS-4250-XL

Some IDS-4250-XLs were shipped with faulty DIMMs on the XL cards. The faulty DIMMs cause the sensor to hang or SensorApp to stop functioning and generate a core file.

For the procedure for checking IDS-4250-XL for faulty memory, see the Partner Field Notice 52563.

Blocking

This section provides troubleshooting help for blocking and the ARC service. It contains the following topics.

Troubleshooting Blocking

After you have configured ARC, you can verify if it is running properly by using the **show version** command. To verify that ARC is connecting to the network devices, use the **show statistics network-access** command.

Note ARC was formerly known as Network Access Controller. Although the name has been changed for IPS 5.1, it still appears in IDM and the CLI as Network Access Controller, **nac**, and **network-access**.

To troubleshoot ARC, follow these steps:

1. Verify that ARC is running.

   For the procedure, see Verifying ARC is Running, page C-20.
2. Verify that ARC is connecting to the network devices.

   For the procedure, see Verifying ARC Connections are Active, page C-21.
3. Verify that the Event Action is set to Block Host for specific signatures.

   For the procedure, see Blocking Not Occurring for a Signature, page C-25.
4. Verify that the master blocking sensor is properly configured.

   For the procedure, see Verifying the Master Blocking Sensor Configuration, page C-26.

Note For a discussion of ARC architecture, see Attack Response Controller, page A-11.

Verifying ARC is Running

To verify that ARC is running, use the **show version** command. If MainApp is not running, ARC cannot run. ARC is part of MainApp.

To verify ARC is running, follow these steps:

Step 1 Log in to the CLI.

Step 2 Verify that MainApp is running:

```
sensor# show version
Application Partition:

Cisco Intrusion Prevention System, Version 5.0(1.1)S152.0

OS Version 2.4.26-IDS-smp-bigphys
Platform: IPS-4255-K9
Serial Number: JAB0815R017
No license present
```

```
Sensor up-time is 3 days.
Using 734863360 out of 3974291456 bytes of available memory (18% usage)
system is using 17.3M out of 29.0M bytes of available disk space (59% usage)
application-data is using 35.6M out of 166.8M bytes of available disk space (23% usage)
boot is using 40.5M out of 68.6M bytes of available disk space (62% usage)


MainApp          2005_Mar_04_14.23   (Release)   2005-03-04T14:35:11-0600   Not Running
AnalysisEngine   2005_Mar_18_12.53   (Release)   2005-03-18T13:03:21-0600   Running
CLI              2005_Mar_04_14.23   (Release)   2005-03-04T14:35:11-0600


Upgrade History:

  IDS-K9-sp-5.0-1.1-   12:53:00 UTC Fri Mar 18 2005

Recovery Partition Version 1.1 - 5.0(1.1)

sensor#
```

Step 3 If MainApp displays `Not Running`, ARC has failed. Contact the TAC.

Verifying ARC Connections are Active

If the State is not `Active` in the ARC statistics, there is a problem.

To verify that the State is Active in the statistics, follow these steps:

Step 1 Log in to the CLI.

Step 2 Verify that ARC is connecting:

Check the State section of the output to verify that all devices are connecting.

```
sensor# show statistics network-access
Current Configuration
   LogAllBlockEventsAndSensors = true
   EnableNvramWrite = false
   EnableAclLogging = false
   AllowSensorBlock = false
   BlockMaxEntries = 250
   MaxDeviceInterfaces = 250
   NetDevice
      Type = Cisco
      IP = 10.89.147.54
      NATAddr = 0.0.0.0
      Communications = telnet
      BlockInterface
         InterfaceName = fa0/0
         InterfaceDirection = in
State
   BlockEnable = true
   NetDevice
      IP = 10.89.147.54
      AclSupport = uses Named ACLs
      Version = 12.2
      State = Active
sensor#
```

Step 3 If ARC is not connecting, look for recurring errors:

```
sensor# show events error hh:mm:ss month day year | include : nac
```

Example:

```
sensor# show events error 00:00:00 Apr 01 2005 | include : nac
```

Step 4 Make sure you have the latest software updates:

```
sensor# show version
Upgrade History:

  IDS-K9-maj-5.0-1-   14:16:00 UTC Thu Mar 04 2004

Recovery Partition Version 1.1 - 5.0(1)S149
```

If you do not have the latest software updates, download them from Cisco.com. For the procedure, see Obtaining Cisco IPS Software, page 18-1.

Step 5 Read the Readme that accompanies the software upgrade for any known DDTS for ARC.

Step 6 Make sure the configuration settings for each device are correct (the username, password, and IP address). For the procedure, see Device Access Issues, page C-22.

Step 7 Make sure the interface and directions for each network device are correct. For the procedure, see Verifying the Interfaces and Directions on the Network Device, page C-24.

Step 8 If the network device is using SSH-DES or SSH-3DES, make sure that you have enabled SSH connections to the device. For the procedure, see Enabling SSH Connections to the Network Device, page C-24.

Step 9 Verify that each interface and direction on each controlled device is correct. For the procedure, see Verifying the Interfaces and Directions on the Network Device, page C-24.

Device Access Issues

ARC may not be able to access the devices it is managing. Make sure the you have the correct IP address and username and password for the managed devices and the correct interface and direction configured.

Note SSH devices must support SSH 1.5. The sensor does not support SSH 2.0.

To troubleshoot device access issues, follow these steps:

Step 1 Log in to the CLI.

Step 2 Verify the IP address for the managed devices:

```
sensor# configure terminal
sensor (config)# service network-access
sensor(config-net)# show settings
   general
   ----------------------------------------------
      log-all-block-events-and-errors: true <defaulted>
      enable-nvram-write: false <defaulted>
      enable-acl-logging: false <defaulted>
      allow-sensor-block: false <defaulted>
      block-enable: true <defaulted>
      block-max-entries: 250 <defaulted>
```

```
      max-interfaces: 250 <defaulted>
      master-blocking-sensors (min: 0, max: 100, current: 0)
      ----------------------------------------------
      ----------------------------------------------
      never-block-hosts (min: 0, max: 250, current: 0)
      ----------------------------------------------
      ----------------------------------------------
      never-block-networks (min: 0, max: 250, current: 0)
      ----------------------------------------------
      ----------------------------------------------
      block-hosts (min: 0, max: 250, current: 0)
      ----------------------------------------------
      ----------------------------------------------
      block-networks (min: 0, max: 250, current: 0)
      ----------------------------------------------
      ----------------------------------------------
   ---------------------------------------------
   user-profiles (min: 0, max: 250, current: 1)
   ---------------------------------------------
      profile-name: r7200
      ----------------------------------------------
         enable-password: <hidden>
         password: <hidden>
         username: netrangr default:
      ----------------------------------------------
   ---------------------------------------------
   cat6k-devices (min: 0, max: 250, current: 0)
   ---------------------------------------------
   ---------------------------------------------
   router-devices (min: 0, max: 250, current: 1)
   ---------------------------------------------
      ip-address: 10.89.147.54
      ----------------------------------------------
         communication: telnet default: ssh-3des
         nat-address: 0.0.0.0 <defaulted>
         profile-name: r7200
         block-interfaces (min: 0, max: 100, current: 1)
         ----------------------------------------------
            interface-name: fa0/0
            direction: in
            ----------------------------------------------
               pre-acl-name: <defaulted>
               post-acl-name: <defaulted>
            ----------------------------------------------
         ----------------------------------------------
      ----------------------------------------------
   ---------------------------------------------
   firewall-devices (min: 0, max: 250, current: 0)
   ---------------------------------------------
   ---------------------------------------------
sensor(config-net)#
```

Step 3 Manually connect to the device to make sure you have used the correct username, password, and enable password, and to ensure that the device is reachable from the sensor.

a. Log in to the service account.

b. Telnet or SSH to the network device to verify the configuration.

c. Make sure you can reach the device.

d. Verify the username and password.

Step 4 Verify that each interface/direction on each network device is correct. For the procedure, see Verifying the Interfaces and Directions on the Network Device, page C-24.

Verifying the Interfaces and Directions on the Network Device

To verify that each interface and direction on each controlled device is correct, you can send a manual block to a bogus host and then check to see if deny entries exist for the blocked addresses in the ACL of the router.

Note Click **Monitoring > Active Host Blocks** to perform a manual block from IDM.

To initiate a manual block to a bogus host, follow these steps:

Step 1 Enter ARC general submode:

```
sensor# configure terminal
sensor(config)# service network-access
sensor(config-net)# general
```

Step 2 Start the manual block of the bogus host IP address:

```
sensor(config-net-gen)# block-hosts 10.16.0.0
```

Step 3 Exit general submode:

```
sensor(config-net-gen)# exit
sensor(config-net)# exit
Apply Changes:? [yes]:
```

Step 4 Press **Enter** to apply the changes or type **no** to discard them.

Step 5 Telnet to the router and verify that a deny entry for the blocked address exists in the ACL of the router.

Refer to the router documentation for the procedure.

Step 6 Remove the manual block by repeating Steps 1 through 4 except in Step 2 place **no** in front of the command:

```
sensor(config-net-gen)# no block-hosts 10.16.0.0
```

Enabling SSH Connections to the Network Device

If you are using SSH-DES or SSH-3DES as the communication protocol for the network device, you must make sure you have enabled it on the device.

To enable SSH connections to the network device, follow these steps:

Step 1 Log in to the CLI.

Step 2 Enter configuration mode:

```
sensor# configure terminal
```

Step 3 Enable SSH:

```
sensor(config)# ssh host blocking_device_ip_ address
```

Step 4 Type **yes** when prompted to accept the device.

Blocking Not Occurring for a Signature

If blocking is not occurring for a specific signature, check that the event action is set to block the host.

To make sure blocking is occurring for a specific signature, follow these steps:

Step 1 Log in to the CLI.

Step 2 Enter signature definition submode:

```
sensor# configure terminal
sensor(config)# service signature-definition sig0
sensor(config-sig)#
```

Step 3 Make sure the event action is set to block the host:

Note If you want to receive alerts, you must always add **produce-alert** any time you configure the event actions.

```
sensor(config-sig)# signatures 1300 0
sensor(config-sig-sig)# engine normalizer
sensor(config-sig-sig-nor)# event-action produce-alert|request-block-host
sensor(config-sig-sig-nor)# show settings
   normalizer
   ----------------------------------------------
      event-action: produce-alert|request-block-host default: produce-alert|deny
-connection-inline
      edit-default-sigs-only
      ----------------------------------------------
         default-signatures-only
         ----------------------------------------------
            specify-service-ports
            ----------------------------------------------
               no
               ----------------------------------------------
               ----------------------------------------------
            ----------------------------------------------
            specify-tcp-max-mss
            ----------------------------------------------
               no
               ----------------------------------------------
               ----------------------------------------------
            ----------------------------------------------
            specify-tcp-min-mss
            ----------------------------------------------
               no
               ----------------------------------------------
               ----------------------------------------------
--MORE--
```

Step 4 Exit signature definition submode:

```
sensor(config-sig-sig-nor)# exit
sensor(config-sig-sig)# exit
sensor(config-sig)# exit
Apply Changes:?[yes]:
```

Step 5 Press **Enter** to apply the changes or type **no** to discard them.

Verifying the Master Blocking Sensor Configuration

To verify that a master blocking sensor is set up properly or to troubleshoot a master blocking sensor that is not set up properly, you can use the **show statistics network-access** command. Make sure that the forwarding sensor is set up as TLS trusted host if the remote master blocking sensor is using TLS for web access.

To verify the master blocking sensor configuration of a sensor, follow these steps:

Step 1 View the ARC statistics and verify that the master blocking sensor entries are in the statistics:

```
sensor# show statistics network-access
Current Configuration
   AllowSensorShun = false
   ShunMaxEntries = 250
   MasterBlockingSensor
      SensorIp = 10.89.149.46
      SensorPort = 443
      UseTls = 1
State
   ShunEnable = true
   ShunnedAddr
      Host
         IP = 122.122.122.44
         ShunMinutes = 60
         MinutesRemaining = 59
```

Step 2 If the master blocking sensor does not show up in the statistics, you need to add it. For the procedure, see Configuring the Sensor to be a Master Blocking Sensor, page 10-28.

Step 3 Initiate a manual block to a bogus host IP address to make sure the master blocking sensor is initialing blocks:

```
sensor# configure terminal
sensor(config)# service network-access
sensor(config-net)# general
sensor(config-net-gen)# block-hosts 10.16.0.0
```

Step 4 Exit network access general submode:

```
sensor(config-net-gen)# exit
sensor(config-net)# exit
Apply Changes:? [yes]:
```

Step 5 Press **Enter** to apply the changes or type **no** to discard them.

Step 6 Verify that the block shows up in the ARC's statistics:

```
sensor# show statistics network-access
Current Configuration
   AllowSensorShun = false
   ShunMaxEntries = 100
```

```
State
   ShunEnable = true
   ShunnedAddr
      Host
         IP = 10.16.0.0
         ShunMinutes =
```

Step 7 Log in to the master blocking sensor host CLI and, using the **show statistics network-access** command, verify that the block also shows up in the master blocking sensor ARC statistics.

```
sensor# show statistics network-access
Current Configuration
   AllowSensorShun = false
   ShunMaxEntries = 250
   MasterBlockingSensor
      SensorIp = 10.89.149.46
      SensorPort = 443
      UseTls = 1
State
   ShunEnable = true
   ShunnedAddr
      Host
         IP = 10.16.0.0
         ShunMinutes = 60
         MinutesRemaining = 59
```

Step 8 If the remote master blocking sensor is using TLS for web access, make sure the forwarding sensor is configured as a TLS host:

```
sensor# configure terminal
sensor(config)# tls trust ip master_blocking_sensor_ip_address
```

Logging

TAC may suggest that you turn on debug logging for troubleshooting purposes. LogApp controls what log messages are generated by each application by controlling the logging severity for different logging zones. By default, debug logging is not turned on.

If you enable individual zone control, each zone uses the level of logging that it is configured for. Otherwise, the same logging level is used for all zones.

This section contains the following topics:

Enabling Debug Logging

Caution Enabling debug logging seriously affects performance and should only be done when instructed by TAC.

To enable debug logging, follow these steps:

Step 1 Log in to the service account.

Step 2 Edit the log.conf file to increase the size of the log to accommodate the additional log statements:

```
vi /usr/cids/idsRoot/etc/log.conf
```

Step 3 Change `fileMaxSizeInK=500` to `fileMaxSizeInK=5000`.

Step 4 Locate the zone and CID section of the file and set the severity to debug:

```
severity=debug
```

Step 5 Save the file, exit the vi editor, and exit the service account.

Step 6 Log in to the CLI as administrator.

Step 7 Enter master control submode:

```
sensor# configure terminal
sensor(config)# service logger
sensor(config-log)# master-control
```

Step 8 To enable debug logging for all zones:

```
sensor(config-log-mas)# enable-debug true
sensor(config-log-mas)# show settings
   master-control
   ----------------------------------------------
      enable-debug: true default: false
      individual-zone-control: false <defaulted>
   ----------------------------------------------
sensor(config-log-mas)#
```

Step 9 To turn on individual zone control:

```
sensor(config-log-mas)# individual-zone-control true
sensor(config-log-mas)# show settings
   master-control
   ----------------------------------------------
      enable-debug: true default: false
      individual-zone-control: true default: false
   ----------------------------------------------
sensor(config-log-mas)#
```

Step 10 Exit master zone control:

```
sensor(config-log-mas)# exit
```

Step 11 View the zone names:

```
sensor(config-log)# show settings
   master-control
   ----------------------------------------------
      enable-debug: false <defaulted>
      individual-zone-control: true default: false
   ----------------------------------------------
   zone-control (min: 0, max: 999999999, current: 14)
   ----------------------------------------------
      <protected entry>
      zone-name: AuthenticationApp
      severity: warning <defaulted>
      <protected entry>
      zone-name: Cid
      severity: debug <defaulted>
```

```
      <protected entry>
      zone-name: Cli
      severity: warning <defaulted>
      <protected entry>
      zone-name: IdapiCtlTrans
      severity: warning <defaulted>
      <protected entry>
      zone-name: IdsEventStore
      severity: warning <defaulted>
      <protected entry>
      zone-name: MpInstaller
      severity: warning <defaulted>
      <protected entry>
      zone-name: cmgr
      severity: warning <defaulted>
      <protected entry>
      zone-name: cplane
      severity: warning <defaulted>
      <protected entry>
      zone-name: csi
      severity: warning <defaulted>
      <protected entry>
      zone-name: ctlTransSource
      severity: warning <defaulted>
      <protected entry>
      zone-name: intfc
      severity: warning <defaulted>
      <protected entry>
      zone-name: nac
      severity: warning <defaulted>
      <protected entry>
      zone-name: sensorApp
      severity: warning <defaulted>
      <protected entry>
      zone-name: tls
      severity: warning <defaulted>
   -----------------------------------------------
sensor(config-log)#
```

For a list of what each zone name refers to, see Zone Names, page C-31.

Step 12 Change the severity level (debug, timing, warning, or error) for a particular zone:

```
sensor(config-log)# zone-control IdsEventStore severity error
sensor(config-log)# show settings
   master-control
   -----------------------------------------------
      enable-debug: true default: false
      individual-zone-control: true default: false
   -----------------------------------------------
   zone-control (min: 0, max: 999999999, current: 14)
   -----------------------------------------------
      <protected entry>
      zone-name: AuthenticationApp
      severity: warning <defaulted>
      <protected entry>
      zone-name: Cid
      severity: debug <defaulted>
      <protected entry>
      zone-name: Cli
      severity: warning <defaulted>
      <protected entry>
      zone-name: IdapiCtlTrans
      severity: warning <defaulted>
```

```
      <protected entry>
      zone-name: IdsEventStore
      severity: error default: warning
      <protected entry>
      zone-name: MpInstaller
      severity: warning <defaulted>
      <protected entry>
      zone-name: cmgr
      severity: warning <defaulted>
      <protected entry>
      zone-name: cplane
      severity: warning <defaulted>
      <protected entry>
      zone-name: csi
      severity: warning <defaulted>
      <protected entry>
      zone-name: ctlTransSource
      severity: warning <defaulted>
      <protected entry>
      zone-name: intfc
      severity: warning <defaulted>
      <protected entry>
      zone-name: nac
      severity: warning <defaulted>
      <protected entry>
      zone-name: sensorApp
      severity: warning <defaulted>
      <protected entry>
      zone-name: tls
      severity: warning <defaulted>
   ----------------------------------------------
sensor(config-log)#
```

Step 13 Turn on debugging for a particular zone:

```
sensor(config-log)# zone-control nac severity debug
sensor(config-log)# show settings
   master-control
   ----------------------------------------------
      enable-debug: true default: false
      individual-zone-control: true default: false
   ----------------------------------------------
   zone-control (min: 0, max: 999999999, current: 14)
   ----------------------------------------------
      <protected entry>
      zone-name: AuthenticationApp
      severity: warning <defaulted>
      <protected entry>
      zone-name: Cid
      severity: debug <defaulted>
      <protected entry>
      zone-name: Cli
      severity: warning <defaulted>
      <protected entry>
      zone-name: IdapiCtlTrans
      severity: warning <defaulted>
      <protected entry>
      zone-name: IdsEventStore
      severity: error default: warning
      <protected entry>
      zone-name: MpInstaller
      severity: warning <defaulted>
      <protected entry>
      zone-name: cmgr
```

```
      severity: warning <defaulted>
      <protected entry>
      zone-name: cplane
      severity: warning <defaulted>
      <protected entry>
      zone-name: csi
      severity: warning <defaulted>
      <protected entry>
      zone-name: ctlTransSource
      severity: warning <defaulted>
      <protected entry>
      zone-name: intfc
      severity: warning <defaulted>
      <protected entry>
      zone-name: nac
      severity: debug default: warning
      <protected entry>
      zone-name: sensorApp
      severity: warning <defaulted>
      <protected entry>
      zone-name: tls
      severity: warning <defaulted>
   ----------------------------------------------
sensor(config-log)#
```

Step 14 Exit the logger submode:

```
sensor(config-log)# exit
Apply Changes:?[yes]:
```

Step 15 Press **Enter** to apply changes or type **no** to discard them:

Zone Names

Table C-1 lists the debug logger zone names:

Table C-1 Debug Logger Zone Names

Zone Name	Description
AuthenticationApp	Authentication zone
Cid	General logging zone
Cli	CLI zone
IdapiCtlTrans	All control transactions zone
IdsEventStore	Event Store zone
MpInstaller	IDSM-2 master partition installer zone
cmgr	Card Manager service zone[1]
cplane	Control Plane zone[2]
csi	CIDS Servlet Interface[3]
ctlTransSource	Outbound control transactions zone
intfc	Interface zone
nac	ARC zone

Table C-1 Debug Logger Zone Names (continued)

Zone Name	Description
sensorApp	AnalysisEngine zone
tls	SSL and TLS zone

1. The Card Manager service is used on AIP-SSM to exchange control and state information between modules in the chassis.
2. The Control Plane is the transport communications layer used by Card Manager on AIP-SSM.
3. The CIDS servlet interface is the interface layer between the CIDS web server and the servlets.

Directing cidLog Messages to SysLog

It might be useful to direct cidLog messages to syslog.

To direct cidLog messages to syslog, follow these steps:

Step 1 Go to the idsRoot/etc/log.conf file.

Step 2 Make the following changes:

a. Set [logApp] `enabled=false`

Comment out the `enabled=true` because `enabled=false` is the default.

b. Set [drain/main] `type=syslog`

The following example shows the logging configuration file:

```
timemode=local
;timemode=utc

[logApp]
;enabled=true
;-------- FIFO parameters --------
fifoName=logAppFifo
fifoSizeInK=240
;-------- logApp zone and drain parameters --------
zoneAndDrainName=logApp
fileName=main.log
fileMaxSizeInK=500

[zone/Cid]
severity=warning
drain=main

[zone/IdsEventStore]
severity=debug
drain=main

[drain/main]
type=syslog
```

The syslog output is sent to the syslog facility local6 with the following correspondence to syslog message priorities:

```
LOG_DEBUG,        //   debug
LOG_INFO,             //    timing
LOG_WARNING,   //    warning
```

```
LOG_ERR,              //    error
LOG_CRIT              //     fatal
```

Note Make sure that your /etc/syslog.conf has that facility enabled at the proper priority.

Caution The syslog is much slower than logApp (about 50 messages per second as opposed to 1000 or so). We recommend that you enable debug severity on one zone at a time.

TCP Reset Not Occurring for a Signature

If you do not have the event action set to reset, the TCP reset does not occur for a specific signature.

To troubleshoot a reset not occurring for a specific signature, follow these steps:

Step 1 Log in to the CLI.

Step 2 Make sure the event action is set to TCP reset:

```
sensor# configure terminal
sensor(config)# service signature-definition sig0
sensor(config-sig)# signatures 1000 0
sensor(config-sig-sig)# engine atomic-ip
sensor(config-sig-sig-ato)# event-action reset-tcp-connection|produc-alert
sensor(config-sig-sig-ato)# show settings
   atomic-ip
   ----------------------------------------------
      event-action: produce-alert|reset-tcp-connection default: produce-alert
      fragment-status: any <defaulted>
      specify-l4-protocol
      ----------------------------------------------
         no
         ----------------------------------------------
         ----------------------------------------------
      ----------------------------------------------
      specify-ip-payload-length
      ----------------------------------------------
         no
         ----------------------------------------------
         ----------------------------------------------
      ----------------------------------------------
      specify-ip-header-length
      ----------------------------------------------
         no
         ----------------------------------------------
         ----------------------------------------------
      ----------------------------------------------
      specify-ip-tos
      ----------------------------------------------
--MORE--
```

Step 3 Exit signature definition submode:

```
sensor(config-sig-sig-ato)# exit
sensor(config-sig-sig)# exit
sensor(config-sig)# exit
Apply Changes:?[yes]:
```

Step 4 Press **Enter** to apply the changes or type **no** to discard them.

Step 5 Make sure the correct alerts are being generated:

```
sensor# show events alert
evAlert: eventId=1047575239898467370 severity=medium
originator:
hostId: sj_4250_40
appName: sensorApp
appInstanceId: 1004
signature: sigId=20000 sigName=STRING.TCP subSigId=0 version=Unknown
addr: locality=OUT 172.16.171.19
port: 32771
victim:
addr: locality=OUT 172.16.171.13 port: 23
actions:
tcpResetSent: true
```

Step 6 Make sure the switch is allowing incoming TCP reset packet from the sensor.

Refer to your switch documentation for the procedure.

Step 7 Make sure the resets are being sent:

```
root# ./tcpdump -i eth0 src host 172.16.171.19
tcpdump: WARNING: eth0: no IPv4 address assigned
tcpdump: listening on eth0
13:58:03.823929 172.16.171.19.32770 > 172.16.171.13.telnet: R 79:79(0) ack 62 win 0
13:58:03.823930 172.16.171.19.32770 > 172.16.171.13.telnet: R 80:80(0) ack 62 win 0
13:58:03.823930 172.16.171.19.32770 > 172.16.171.13.telnet: R 80:80(0) ack 62 win 0
13:58:03.823930 172.16.171.19.32770 > 172.16.171.13.telnet: R 80:80(0) ack 62 win 0
```

Software Upgrades

This section helps in troubleshooting software upgrades. It contains the following topics:

IDS-4235 and IDS-4250 Hang During A Software Upgrade

If the BIOS of IDS-4235 and IDS-4250 is at A03, you must upgrade it to A04 before applying the most recent IPS software, otherwise, the appliances hang during the software upgrade process. For the procedure for upgrading the BIOS, refer to Upgrading the BIOS. For the procedure for applying the latest IPS software, see Obtaining Cisco IPS Software, page 18-1.

Which Updates to Apply and Their Prerequisites

You must have the correct service pack and minor and major version of the software. If you are having trouble with applying new software, make sure that you are applying the proper updates with the proper prerequisites. For more information on software file versioning, see IPS Software Versioning, page 18-3.

Issues With Automatic Update

The following list provides suggestions for troubleshooting automatic update:

- Run tcpDump
  - Create a service account. Su to root and run tcpDump on the command and control interface to capture packets between the sensor and the FTP server.

    For the procedure, see Creating the Service Account, page 4-13.
  - Use the **upgrade** command to manually upgrade the sensor.

    For the procedure, see Chapter 17, "Upgrading, Downgrading, and Installing System Images."
  - Look at the tcpDump output for errors coming back from the FTP server.
- Make sure the sensor is in the correct directory.

  The directory must be specified correctly. This has caused issues with Windows FTP servers. Sometimes an extra "/" or even two "/" are needed in front of the directory name.

  To verify this, use the same FTP commands you see in the tcpDump output through your own FTP connection.
- Make sure you have not modified the FTP server to use custom prompts.

  If you modify the FTP prompts to give security warnings, for example, this causes a problem, because the sensor is expecting a hard-coded list of responses.

  Note Not modifying the prompt only applies to versions before 4.1(4).
- You must use the Windows FTP server setup option to emulate UNIX file structure and not MS-DOS file structure.
- If you are using SCP, make sure you have added the SSH host key to the known hosts list.

  For the procedure, see Adding Hosts to the Known Hosts List, page 4-32.

Try the manual **upgrade** command before attempting the automatic update. If it works with the **upgrade** command and does not work with the automatic update, try the following:

- Determine which IPS software version your sensor has (for the procedure, see Displaying Version Information, page 13-19.

  Version 4.0(1) has a known problem with automatic update. Upgrade manually to 4.1(1) before trying to configure and use automatic update.
- Make sure the passwords configured for automatic update. Make sure they match the same passwords used for manual update.
- Make sure that the filenames in the FTP server are exactly what you see on Downloads on Cisco.com. This includes capitalization.

  Some Windows FTP servers allow access to the file with the incorrect capitalization but the sensor ultimately rejects the file because the name has changed.

If necessary, run tcpDump on automatic update. You can compare the successful manual update with the unsuccessful automatic update and troubleshoot from there.

Updating a Sensor with the Update Stored on the Sensor

You can store the update package in the /var directory on the sensor and update the sensor from there if you need to.

To update the sensor with an update stored on the sensor, follow these steps:

Step 1 Log in to the service account.

Step 2 Obtain the update package file from Cisco.com.

For the procedure, see Obtaining Cisco IPS Software, page 18-1.

Step 3 FTP or SCP the update file to the sensor's /usr/cids/idsRoot/var directory.

Step 4 Set the file permissions:

```
chmod 644 ips_package_file_name
```

Step 5 Exit the service account.

Step 6 Log in to the sensor using an account with administrator privileges.

Step 7 Store the sensor's host key:

```
sensor# configure terminal
sensor(config)# service ssh
sensor(config-ssh)# rsa1-keys sensor_ip_address
```

Step 8 Upgrade the sensor:

```
sensor(config)# upgrade scp://service@sensor_ip_address/upgrade/ips_package_file_name
Enter password: *****
Re-enter password: *****
```

UNIX-Style Directory Listings

To configure Auto Update using an FTP server, the FTP server must provide directory listing responses in UNIX style. MS-DOS style directory listing is not supported by the sensor Auto Update feature.

Note If the server supplies MS-DOS style directory listings, the sensor cannot parse the directory listing and does not know that there is a new update available.

To change Microsoft IIS to use UNIX-style directory listings, follow these steps:

Step 1 **Choose Start > Program Files > Administrative Tools**.

Step 2 Click the **Home Directory** tab.

Step 3 Click the **UNIX directory listings style** radio button.

Troubleshooting IDM

Note These procedures also apply to the IPS section of ASDM.

Note After you upgrade any IPS software on your sensor, you must restart the IDM to see the latest software features.

This section contains troubleshooting procedures for IDM. This section contains the following topics:

Increasing the Memory Size of the Java Plug-In

To correctly run IDM, your browser must have Java Plug-in 1.4.2 or 1.5 installed. By default the Java Plug-in allocates 64 MB of memory to IDM. IDM can run out of memory while in use, which can cause IDM to freeze or display blank screens. Running out of memory can also occur when you click **Refresh**. An `OutofMemoryError` message appears in the Java console whenever this occurs.

Note We recommend that you use Sun Microsystems Java. Using any other version of Java could cause problems with IDM.

You must change the memory settings of Java Plug-in before using IDM. The mandatory minimum memory size is 256 MB.

This section contains the following topics:

Java Plug-In on Windows

To change the settings of Java Plug-in on Windows for Java Plug-in 1.4.2 and 1.5, follow these steps:

Step 1 Close all instances of Internet Explorer or Netscape.

Step 2 Click **Start > Settings > Control Panel**.

Step 3 If you have Java Plug-in 1.4.2 installed:

a. Click Java Plug-in.

The Java Plug-in Control Panel appears.

b. Click the **Advanced** tab.

c. Type `-Xms256m` in the **Java RunTime Parameters** field.

d. Click **Apply** and exit the Java Control Panel.

Step 4 If you have Java Plug-in 1.5 installed:

a. Click Java.

The Java Control Panel appears.

b. Click the **Java** tab.

c. Click **View** under Java Applet Runtime Settings.

The Java Runtime Settings Panel appears.

d. Type `-Xms256m` in the **Java Runtime Parameters** field and then click **OK**.

e. Click **OK** and exit the Java Control Panel.

Java Plug-In on Linux and Solaris

To change the settings of Java Plug-in 1.4.2 or 1.5 on Linux and Solaris, follow these steps:

Step 1 Close all instances of Netscape or Mozilla.

Step 2 Bring up Java Plug-in Control Panel by launching the ControlPanel executable file.

Note In the Java 2 SDK, this file is located at <SDK installation directory>/jre/bin/ControlPanel. For example if your Java 2 SDK is installed at /usr/j2se, the full path is /usr/j2se/jre/bin/ControlPanel.

Note In a Java 2 Runtime Environment installation, the file is located at <JRE installation directory>/bin/ControlPanel.

Step 3 If you have Java Plug-in 1.4.2 installed:

a. Click the **Advanced** tab.

b. Type `-Xms256m` in the **Java RunTime Parameters** field.

c. Click **Apply** and close the Java Control Panel.

Step 4 If you have Java Plug-in 1.5 installed:

a. Click the **Java** tab.

b. Click **View** under Java Applet Runtime Settings.

c. Type `-Xms256m` in the Java Runtime Parameters field and then click **OK**.

d. Click **OK** and exit the Java Control Panel.

Cannot Launch IDM - Loading Java Applet Failed

Symptom The browser displays `Loading Cisco IDM. Please wait ...` At the bottom left corner of the window, `Loading Java Applet Failed` is displayed.

Possible Cause This condition can occur if multiple Java Plug-ins (1.4.*x* and/or 1.3.*x*) are installed on the machine on which you are launching the IDM.

Recommended Action Clear the Java cache and remove temp files and clear history in the browser you are using. The result is that neither of these plug-ins will be used by default and each applet should use the correct plug-in.

To clear the cache, follow these steps:

Step 1 Close all browser windows.

Step 2 If you have Java Plug-in 1.3.*x* installed:

a. Click **Start** > **Settings** > **Control Panel** > **Java Plug-in 1.3.x**.
b. Click the **Advanced** tab.
c. Under Java Runtime Environment, select **JRE 1.3.x** from the drop-down menu.
d. Click the **Cache** tab.
e. Click **Clear**.

Step 3 If you have Java Plug-in 1.4.*x* installed:

a. Click **Start** > **Settings** > **Control Panel** > **Java Plug-in 1.4.x**.
b. Click the **Advanced** tab.
c. Under Java Runtime Environment, select **JRE 1.3.x** from the drop-down menu.
d. Click the **Cache** tab.
e. Click the **Browser** tab.
f. Deselect all browser check boxes.
g. Click **Clear Cache**.

Step 4 Delete the temp files and clear the history in the browser.

Cannot Launch IDM -Analysis Engine Busy

Error Message `Error connecting to sensor. Failed to load sensor-errNotAvailable-Analysis Engine is busy. Exiting IDM.`

Possible Cause This condition can occur if the Analysis Engine in the sensor is busy getting ready to perform a task and so does not respond to IDM.

Recommended Action Wait for a while and try again to connect.

IDM, Remote Manager, or Sensing Interfaces Cannot Access the Sensor

If IDM, a remote manager, or sensing interfaces cannot access the sensor, but you can access the sensor's CLI using SSH or Telnet (if enabled), follow these steps:

Note For the procedure for enabling and disabling Telnet on the sensor, see Enabling and Disabling Telnet, page 4-4.

Step 1 Make sure the network configuration allows access to the web server port that is configured on the sensor:

```
sensor# setup


    --- System Configuration Dialog ---

At any point you may enter a question mark '?' for help.
User ctrl-c to abort configuration dialog at any prompt.
Default settings are in square brackets '[]'.


Current Configuration:


service host
network-settings
host-ip 10.89.130.108/23,10.89.130.1
host-name sensor
telnet-option enabled
access-list 0.0.0.0/0
ftp-timeout 300
no login-banner-text
exit
time-zone-settings
offset 0
standard-time-zone-name UTC
exit
summertime-option disabled
ntp-option disabled
exit
service web-server
port 443
exit
```

For more information, see Changing Web Server Settings, page 4-9.

Step 2 If network devices, such as routers, switches, or firewalls, are between the sensor and the workstation, make sure these devices are configured to allow the workstation to access the sensor's web server port.

All remote management communication is performed by the sensor's web server.

For more information, see Changing Web Server Settings, page 4-9.

Signatures Not Producing Alerts

If you are not seeing any alerts when signatures are firing, make sure that you have configured Produce Alert as an event action.

Caution You cannot add other actions each time you configure the event actions. You are actually replacing the list of event actions every time you configure it, so make sure you choose Produce Alert every time you configure event actions.

For example, if you choose Produce Alert, but later add another event action and do not add Produce Alert to the new configuration, alerts will not be sent to the Event Store. To make sure you are getting alerts, use statistics for the virtual sensor and event store.

Troubleshooting IDSM-2

IDSM-2 has the same software architecture as the 4200 series sensors. You can use the same troubleshooting tools as outlined in Troubleshooting the 4200 Series Appliance, page C-8.

This section pertains specifically to troubleshooting IDSM-2.

This section contains the following topics:

Diagnosing IDSM-2 Problems

Use the following list to diagnose IDSM-2 problems:

- The ribbon cable between IDSM-2 and the motherboard is loose.

  During physical handling of the module, the connector can come loose from the base card, and cause the daughter card and the base card to lose contact with each other. A loose ribbon cable connector causes an on-line diagnostic error on ports 7 and 8. The module cannot operate when this condition exists. For more information, see Partner Field Notice 52816.
- Some IDSM-2s were shipped with faulty DIMMs. For the procedure for checking IDSM-2 for faulty memory see the Partner Field 52563.
- The hard-disk drive fails to read or write. When the hard-disk drive has been in constant use for extended periods of time (for more than 2 weeks), multiple symptoms, such as the following, can occur:
  - An inability to log in
  - I/O errors to the console when doing read/write operations (the **ls** command)

  - Commands do not execute properly (cannot find the path to the executable)

  The switch reports that the module is ok, but if you log in to the Service account and try to execute commands, you see that the problem exists. The 4.1(4) service pack alleviates this problem, but if you reimage IDSM-2 with the 4.1(4) application partition image, you must apply the 4.1(4b) patch. For more information see CSCef12198.

- SensorApp either crashes or takes 99% of the CPU when IP logging is enabled for stream-based signatures (1300 series). For the workaround, see CSCed32093.
- IDSM-2 appears to lock up and remote access is prohibited (SSH, Telnet, IDM, Event Server, Control Transaction Server, and IP log Server). This defect is related to using SWAP. IDSM-2 responds to pings. Apply the 4.1(4) service pack to resolve this issue. For more information, see CSCed54146.
- Shortly after you upgrade IDSM-2 or you tune a signature with VMS, IDSM-2 becomes unresponsive and often produces a SensorApp core file. Apply the 4.1(4b) patch to fix this issue.
- Confirm that IDSM-2 has the supported configurations. For more information, see Minimum Supported IDSM-2 Configurations, page 15-4.

If you have confirmed that IDSM-2 does not suffer from any of the problems listed above and yet it appears unresponsive, for example, you cannot log in through SSH or Telnet, nor can you session to the switch, determine if IDSM-2 responds to pings and if you can log in through the service account. If you can log in, obtain a cidDump and any core files and contact TAC.

Switch Commands for Troubleshooting

The following switch commands help you troubleshoot IDSM-2:

- **show module** (Cisco Catalyst Software and Cisco IOS Software)
- **show version** (Cisco Catalyst Software and Cisco IOS Software)
- **show port** (Cisco Catalyst Software)
- **show trunk** (Cisco Catalyst Software)
- **show span** (Cisco Catalyst Software)
- **show security acl** (Cisco Catalyst Software)
- **show intrusion-detection module** (Cisco IOS Software)
- **show monitor** (Cisco IOS Software)
- **show vlan access-map** (Cisco IOS Software)
- **show vlan filter** (Cisco IOS Software)

Status LED Off

If the status indicator is off on IDSM-2, you need to turn power on to IDSM-2.

To determine status of IDSM-2, follow these steps:

Step 1 Log in to the console.

Step 2 Verify that IDSM-2 is online:

For Catalyst Software:

```
cat6k> enable

Enter password:
cat6k> (enable) show module
Mod Slot Ports Module-Type                Model               Sub Status
--- ---- ----- ------------------------- ------------------- --- --------
1   1    2     1000BaseX Supervisor      WS-X6K-SUP1A-2GE    yes ok
15  1    1     Multilayer Switch Feature WS-F6K-MSFC         no  ok
2   2    48    10/100BaseTX Ethernet     WS-X6248-RJ-45      no  ok
3   3    48    10/100/1000BaseT Ethernet WS-X6548-GE-TX      no  ok
4   4    16    1000BaseX Ethernet        WS-X6516A-GBIC      no  ok
6   6    8     Intrusion Detection Mod   WS-SVC-IDSM2        yes ok

Mod Module-Name          Serial-Num
--- -------------------- -----------
1                        SAD041308AN
15                       SAD04120BRB
2                        SAD03475400
3                        SAD073906RC
4                        SAL0751QYN0
6                        SAD062004LV

Mod MAC-Address(es)                         Hw     Fw          Sw
--- -------------------------------------- ------ ---------- ----------------
1   00-d0-c0-cc-0e-d2 to 00-d0-c0-cc-0e-d3 3.1    5.3.1      8.4(1)
    00-d0-c0-cc-0e-d0 to 00-d0-c0-cc-0e-d1
    00-30-71-34-10-00 to 00-30-71-34-13-ff
15  00-30-7b-91-77-b0 to 00-30-7b-91-77-ef 1.4    12.1(23)E2 12.1(23)E2
2   00-30-96-2b-c7-2c to 00-30-96-2b-c7-5b 1.1    4.2(0.24)V 8.4(1)
3   00-0d-29-f6-01-98 to 00-0d-29-f6-01-c7 5.0    7.2(1)     8.4(1)
4   00-0e-83-af-15-48 to 00-0e-83-af-15-57 1.0    7.2(1)     8.4(1)
6   00-e0-b0-ff-3b-80 to 00-e0-b0-ff-3b-87 0.102  7.2(0.67)  5.0(0.30)

Mod Sub-Type                Sub-Model           Sub-Serial  Sub-Hw Sub-Sw
--- ----------------------- ------------------- ----------- ------ ------
1   L3 Switching Engine     WS-F6K-PFC          SAD041303G6 1.1
6   IDS 2 accelerator board WS-SVC-IDSUPG       .           2.0
cat6k> (enable)
```

For Cisco IOS software:

```
switch#show module
Mod Ports Card Type                                Model              Serial No.
--- ----- ------------------------------------- ------------------ -----------
  1   48  48 port 10/100 mb RJ-45 ethernet       WS-X6248-RJ-45     SAD0401012S
  2   48  48 port 10/100 mb RJ45                 WS-X6348-RJ-45     SAL04483QBL
  3   48  SFM-capable 48 port 10/100/1000mb RJ45 WS-X6548-GE-TX     SAD073906GH
  5    8  Intrusion Detection System             WS-SVC-IDSM-2      SAD0751059U
  6   16  SFM-capable 16 port 1000mb GBIC        WS-X6516A-GBIC     SAL0740MMYJ
  7    2  Supervisor Engine 720 (Active)         WS-SUP720-3BXL     SAD08320L2T
  9    1  1 port 10-Gigabit Ethernet Module      WS-X6502-10GE      SAD071903BT
```

```
 11    8  Intrusion Detection System            WS-SVC-IDSM2       SAD05380608
 13    8  Intrusion Detection System            WS-SVC-IDSM-2      SAD072405D8

Mod MAC addresses                       Hw    Fw           Sw           Status
--- --------------------------------- ------ ------------ ------------ -------
  1  00d0.d328.e2ac to 00d0.d328.e2db   1.1   4.2(0.24)VAI 8.5(0.46)ROC Ok
  2  0003.6c14.e1d0 to 0003.6c14.e1ff   1.4   5.4(2)       8.5(0.46)ROC Ok
  3  000d.29f6.7a80 to 000d.29f6.7aaf   5.0   7.2(1)       8.5(0.46)ROC Ok
  5  0003.fead.651a to 0003.fead.6521   4.0   7.2(1)       5.0(1.1)     Ok
  6  000d.ed23.1658 to 000d.ed23.1667   1.0   7.2(1)       8.5(0.46)ROC Ok
  7  0011.21a1.1398 to 0011.21a1.139b   4.0   8.1(3)       12.2(PIKESPE Ok
  9  000d.29c1.41bc to 000d.29c1.41bc   1.3   Unknown      Unknown      PwrDown
 11  00e0.b0ff.3340 to 00e0.b0ff.3347   0.102 7.2(0.67)    5.0(1.1)     Ok
 13  0003.feab.c850 to 0003.feab.c857   4.0   7.2(1)       5.0(1)       Ok

Mod Sub-Module                  Model              Serial        Hw     Status
--- --------------------------- ------------------ ------------ ------- -------
  5 IDS 2 accelerator board     WS-SVC-IDSUPG      07E91E508A    2.0    Ok
  7 Policy Feature Card 3       WS-F6K-PFC3BXL     SAD083305A1   1.3    Ok
  7 MSFC3 Daughterboard         WS-SUP720          SAD083206JX   2.1    Ok
 11 IDS 2 accelerator board     WS-SVC-IDSUPG      .             2.0    Ok
 13 IDS 2 accelerator board     WS-SVC-IDSUPG      0347331976    2.0    Ok

Mod Online Diag Status
--- -------------------
  1 Pass
  2 Pass
  3 Pass
  5 Pass
  6 Pass
  7 Pass
  9 Unknown
 11 Pass
 13 Pass
switch#
```

Note It is normal for the status to read `other` when IDSM-2 is first installed. After IDSM-2 completes the diagnostics routines and comes online, the status reads `ok`. Allow up to 5 minutes for IDSM-2 to come online.

Step 3 If the status does not read `ok`, turn the module on:

```
switch# set module power up module_number
```

Status LED On But IDSM-2 Does Not Come Online

If the status indicator is on, but IDSM-2 does not come online, try the following troubleshooting tips:

- Reset IDSM-2.
- Make sure IDSM-2 is installed properly in the switch.
- If the hard-disk drive status has failed, reimage the application partition.

To enable IDSM-2, follow these steps:

Step 1 Log in to the console.

Step 2 Make sure IDSM-2 is enabled:

```
router# show module
```

Step 3 If the status does not read `ok`, enable IDSM-2:

```
router# set module enable module_number
```

Step 4 If IDSM-2 still does not come online, reset it:

```
router# reset module_number
```

Wait for about 5 minutes for IDSM-2 to come online.

Step 5 If IDSM-2 still does not come online, make sure the hardware and operating system are ok:

```
router# show test module_number
```

Step 6 If the `port` status reads `fail`, make sure IDSM-2 is firmly connected in the switch.

Step 7 If the `hdd` status reads `fail`, you must reimage the application partition.

For the procedure, see Chapter 17, "Upgrading, Downgrading, and Installing System Images."

Cannot Communicate With IDSM-2 Command and Control Port

If you cannot communicate with the IDSM-2 command and control port, the command and control port may not be in the correct VLAN.

To communicate with the command and control port of IDSM-2, follow these steps:

Step 1 Log in to the console.

Step 2 Make sure you can ping the command port from any other system.

Step 3 Make sure the IP address, mask, and gateway settings are correct:

```
router# show configuration
```

Step 4 Make sure the command and control port is in the correct VLAN:

For Catalyst software:

```
cat6k> (enable) show port 6/8
* = Configured MAC Address

# = 802.1X Authenticated Port Name.

Port  Name                 Status     Vlan       Duplex Speed       Type
----- -------------------- ---------- ---------- ------ ----------- ------------
 6/8                       connected  trunk        full        1000 IDS


Port  Status      ErrDisable Reason    Port ErrDisableTimeout  Action on Timeout
----  ----------  -------------------  ----------------------  -----------------
 6/8  connected                      -  Enable                  No Change
```

```
Port  Align-Err  FCS-Err    Xmit-Err   Rcv-Err    UnderSize
----- ---------- ---------- ---------- ---------- ---------
 6/8           0          0          0          0         0

Port  Single-Col Multi-Coll Late-Coll  Excess-Col Carri-Sen Runts     Giants
----- ---------- ---------- ---------- ---------- --------- --------- ---------
 6/8           0          0          0          0         0         0         -

Port  Last-Time-Cleared
----- --------------------------
 6/8  Wed Mar 2 2005, 15:29:49

Idle Detection
--------------
   --
cat6k> (enable)
```

For Cisco IOS software:

```
cat6k#show intrusion-detection module 5 management-port state
Intrusion-detection module 5 management-port:

Switchport: Enabled
Administrative Mode: dynamic desirable
Operational Mode: static access
Administrative Trunking Encapsulation: negotiate
Operational Trunking Encapsulation: native
Negotiation of Trunking: On
Access Mode VLAN: 1 (default)
Trunking Native Mode VLAN: 1 (default)
Trunking VLANs Enabled: ALL
Pruning VLANs Enabled: 2-1001
Vlans allowed on trunk:1
Vlans allowed and active in management domain: 1
Vlans in spanning tree forwarding state and not pruned:
   1
Access Vlan = 1


cat6k#
```

Step 5 If the command and control port is not in the correct VLAN, put it in the correct VLAN.

For the procedure, see Configuring the Catalyst 6500 Series Switch for Command and Control Access to IDSM-2, page 15-5.

Using the TCP Reset Interface

IDSM-2 has a TCP reset interface—port 1. IDSM-2 has a specific TCP reset interface because it cannot send TCP resets on its sensing ports.

If you have TCP reset problems with IDSM-2, try the following:

- If the sensing ports are access ports (a single VLAN), you must configure the TCP reset port to be in the same VLAN.
- If the sensing ports are dot1q trunk ports (multi-VLAN), the sensing ports and TCP reset port all must have the same native VLAN, and the TCP reset port must trunk all the VLANs being trunked by both the sensing ports.

Connecting a Serial Cable to IDSM-2

You can connect a serial cable directly to the serial console port on IDSM-2. This lets you bypass the switch and module network interfaces.

To connect a serial cable to IDSM-2, follow these steps:

Step 1 Locate the two RJ-45 ports on IDSM-2.

You can find them approximately in the center of the mother board. If you are facing the module faceplate, the RJ-45 port on the right is the serial console port.

Step 2 Connect a straight-through cable to the right port on IDSM-2, and then connect the other end of the cable to a terminal server port.

Step 3 Configure the terminal server port to be 19200 baud, 8 bits, no parity.

You can now log directly in to IDSM-2.

Note Connecting a serial cable to IDSM-2 works only if there is no module located above IDSM-2 in the switch chassis, because the cable has to come out through the front of the chassis.

Troubleshooting AIP-SSM

AIP-SSM has the same software architecture as the 4200 series sensors. You can use the same troubleshooting tools as outlined in Troubleshooting the 4200 Series Appliance, page C-8.

The following section contains commands that are specific to troubleshooting AIP-SSM.

To see the general health of AIP-SSM, use the **show module 1 details** command:

```
asa# show module 1 details
Getting details from the Service Module, please wait...
ASA 5500 Series Security Services Module-20
Model:              ASA-SSM-20
Hardware version:   0.2
Serial Number:      P2B000005D0
Firmware version:   1.0(10)0
Software version:   5.1(0.1)S153.0
Status:             Up
Mgmt IP addr:       10.89.149.219
Mgmt web ports:     443
Mgmt TLS enabled:   true
asa#
```

The output shows that AIP-SSM is up. If the status reads `Down`, you can reset AIP-SSM using the **hw-module module 1 reset** command:

```
asa# hw-module module 1 reset
The module in slot 1 should be shut down before
resetting it or loss of configuration may occur.
Reset module in slot 1? [confirm]
Reset issued for module in slot 1
asa(config)# show module

Mod Card Type                                    Model              Serial No.
```

```
--- ------------------------------------------ ------------------ -----------
  0 ASA 5520 Adaptive Security Appliance         ASA5520            P2A00000014
  1 ASA 5500 Series Security Services Module-10  ASA-SSM-10         P2A0000067U

Mod MAC Address Range                 Hw Version   Fw Version   Sw Version
--- --------------------------------- ------------ ------------ ---------------
  0 000b.fcf8.7bdc to 000b.fcf8.7be0  0.2          1.0(10)0     7.0(1)
  1 000b.fcf8.0176 to 000b.fcf8.0176  0.2          1.0(10)0     5.1(0.1)S153.0

Mod Status
--- ------------------
  0 Up Sys
  1 Shutting Down
**************************************************
asa(config)# show module

Mod Card Type                                     Model              Serial No.
--- ------------------------------------------ ------------------ -----------
  0 ASA 5520 Adaptive Security Appliance         ASA5520            P2A00000014
  1 ASA 5500 Series Security Services Module-10  ASA-SSM-10         P2A0000067U

Mod MAC Address Range                 Hw Version   Fw Version   Sw Version
--- --------------------------------- ------------ ------------ ---------------
  0 000b.fcf8.7bdc to 000b.fcf8.7be0  0.2          1.0(10)0     7.0(1)
  1 000b.fcf8.0176 to 000b.fcf8.0176  0.2          1.0(10)0     5.1(0.1)S153.0

Mod Status
--- ------------------
  0 Up Sys
  1 Up
asa(config)#
```

If you have problems with recovering AIP-SSM, use the **debug module-boot** command to see the output as AIP-SSM boots. Make sure you have the correct IP address for the TFTP server and you have the correct file on the TFTP server. Then use the **hw-module module 1 recover** command again to recover AIP-SSM:

```
asa(config)# hw-module module 1 recover configure
Image URL [tftp://0.0.0.0/]: tftp://10.89.146.1/IPS-SSM-K9-sys-1.1-a-5.1-0.1.i$
Port IP Address [0.0.0.0]: 10.89.150.227
VLAN ID [0]:
Gateway IP Address [0.0.0.0]: 10.89.149.254
asa(config)# debug module-boot
debug module-boot  enabled at level 1
asa(config)# hw-module module 1 recover boot
The module in slot 1 will be recovered. This may erase all configuration and all data on
that device and attempt to download a new image for it.
Recover module in slot 1? [confirm]
Recover issued for module in slot 1
asa(config)# Slot-1 140> Cisco Systems ROMMON Version (1.0(10)0) #0: Fri Mar 25 23:02:10
PST 2005
Slot-1 141> Platform ASA-SSM-10
Slot-1 142> GigabitEthernet0/0
Slot-1 143> Link is UP
Slot-1 144> MAC Address: 000b.fcf8.0176
Slot-1 145> ROMMON Variable Settings:
Slot-1 146>   ADDRESS=10.89.150.227
Slot-1 147>   SERVER=10.89.146.1
Slot-1 148>   GATEWAY=10.89.149.254
Slot-1 149>   PORT=GigabitEthernet0/0
Slot-1 150>   VLAN=untagged
Slot-1 151>   IMAGE=IPS-SSM-K9-sys-1.1-a-5.1-0.1.img
Slot-1 152>   CONFIG=
```

```
Slot-1 153>   LINKTIMEOUT=20
Slot-1 154>   PKTTIMEOUT=4
Slot-1 155>   RETRY=20
Slot-1 156> tftp IPS-SSM-K9-sys-1.1-a-5.1-0.1.img@10.89.146.1 via 10.89.149.254
Slot-1 157> TFTP failure: Packet verify failed after 20 retries
Slot-1 158> Rebooting due to Autoboot error ...
Slot-1 159> Rebooting....
Slot-1 160> Cisco Systems ROMMON Version (1.0(10)0) #0: Fri Mar 25 23:02:10 PST 2005
Slot-1 161> Platform ASA-SSM-10
Slot-1 162> GigabitEthernet0/0
Slot-1 163> Link is UP
Slot-1 164> MAC Address: 000b.fcf8.0176
Slot-1 165> ROMMON Variable Settings:
Slot-1 166>   ADDRESS=10.89.150.227
Slot-1 167>   SERVER=10.89.146.1
Slot-1 168>   GATEWAY=10.89.149.254
Slot-1 169>   PORT=GigabitEthernet0/0
Slot-1 170>   VLAN=untagged
Slot-1 171>   IMAGE=IPS-SSM-K9-sys-1.1-a-5.1-0.1.img
Slot-1 172>   CONFIG=
Slot-1 173>   LINKTIMEOUT=20
Slot-1 174>   PKTTIMEOUT=4
Slot-1 175>   RETRY=20
Slot-1 176> tftp IPS-SSM-K9-sys-1.1-a-5.1-0.1.img@10.89.146.1 via 10.89.149.254
```

Gathering Information

You can use the following CLI commands and scripts to gather information and diagnose the state of the sensor when problems occur. You can use the **show tech-support** command to gather all the sensor's information, or you can use the other individual commands listed in this section for specific information.

This section contains the following topics:

Tech Support Information

The **show tech-support** command is useful for capturing all the status and configuration information of the sensor.

This section describes the **show tech-support** command, and contains the following topics:

Overview

The **show tech-support** command captures all status and configuration information on the sensor and includes the current configuration, version information, and cidDump information. The output can be large, over 1 MB. You can transfer the output to a remote system. for the procedure for copying the output to a remote system, see Displaying Tech Support Information, page C-50.

Note You can get the same information from IDM by clicking Monitoring > Support Information > System Information.

Note Always run the **show tech-support** command before contacting TAC.

Displaying Tech Support Information

Use the **show tech-support** [**page**] [**password**] [**destination-url** *destination-url*] command to display system information on the screen or have it sent to a specific URL. You can use the information as a troubleshooting tool with TAC.

The following parameters are optional:

- **page**—Displays the output, one page of information at a time.

  Press **Enter** to display the next line of output or use the spacebar to display the next page of information.
- **password**—Leaves passwords and other security information in the output.
- **destination-url**—Indicates the information should be formatted as HTML and sent to the destination that follows this command. If you use this keyword, the output is not displayed on the screen.
- *destination-url*—Indicates the information should be formatted as HTML. The URL specifies where the information should be sent. If you do not use this keyword, the information is displayed on the screen.

To display tech support information, follow these steps:

Step 1 Log in to the CLI using an account with administrator privileges.

Step 2 View the output on the screen:

```
sensor# show tech-support page
```

The system information appears on the screen, one page at a time. Press the spacebar to view the next page or press **Ctrl-C** to return to the prompt.

Step 3 To send the output (in HTML format) to a file, follow these steps:

a. Type the following command, followed by a valid destination:

```
sensor# show tech-support destination-url destination-url
```

You can specify the following destination types:

- **ftp:**—Destination URL for FTP network server. The syntax for this prefix is `ftp:[[//username@location]/relativeDirectory]/filename` or `ftp:[[//username@location]//absoluteDirectory]/filename`.
- **scp:**—Destination URL for the SCP network server. The syntax for this prefix is `scp:[[//username@]location]/relativeDirectory]/filename` or `scp:[[//username@]location]//absoluteDirectory]/filename`.

For example, to send the tech support output to the file `/absolute/reports/sensor1Report.html`:

```
sensor# show tech support dest
ftp://csidsuser@10.2.1.2//absolute/reports/sensor1Report.html
```

The `password:` prompt appears.

b. Type the password for this user account.

The `Generating report:` message is displayed.

Tech Support Command Output

The following is an example of the **show tech-support** command output:

Note This output example shows the first part of the command and lists the information for the Interfaces, Network Access Controller, and cidDump services.

```
sensor# show tech-support page

System Status Report
This Report was generated on Fri Feb 21 03:33:52 2003.
Output from show interfaces
Interface Statistics
   Total Packets Received = 0
   Total Bytes Received = 0
   Missed Packet Percentage = 0
   Current Bypass Mode = Auto_off
MAC statistics from interface GigabitEthernet0/1
   Media Type = backplane
   Missed Packet Percentage = 0
   Inline Mode = Unpaired
   Pair Status = N/A
   Link Status = Up
   Link Speed = Auto_1000
   Link Duplex = Auto_Full
   Total Packets Received = 0
   Total Bytes Received = 0
   Total Multicast Packets Received = 0
   Total Broadcast Packets Received = 0
   Total Jumbo Packets Received = 0
   Total Undersize Packets Received = 0
   Total Receive Errors = 0
   Total Receive FIFO Overruns = 0
   Total Packets Transmitted = 0
   Total Bytes Transmitted = 0
   Total Multicast Packets Transmitted = 0
   Total Broadcast Packets Transmitted = 0
   Total Jumbo Packets Transmitted = 0
```

```
   Total Undersize Packets Transmitted = 0
   Total Transmit Errors = 0
   Total Transmit FIFO Overruns = 0
MAC statistics from interface GigabitEthernet0/0
   Media Type = TX
   Link Status = Up
   Link Speed = Auto_100
   Link Duplex = Auto_Full
   Total Packets Received = 2208534
   Total Bytes Received = 157390286
   Total Multicast Packets Received = 20
   Total Receive Errors = 0
   Total Receive FIFO Overruns = 0
   Total Packets Transmitted = 239437
   Total Bytes Transmitted = 107163351
   Total Transmit Errors = 0
   Total Transmit FIFO Overruns = 0

Output from show statistics networkAccess
Current Configuration
   LogAllBlockEventsAndSensors = true
   EnableNvramWrite = false
   EnableAclLogging = false
   AllowSensorBlock = true
   BlockMaxEntries = 250
   MaxDeviceInterfaces = 250
State
   BlockEnable = true

Output from cidDump

cidDiag
CID Diagnostics Report Fri Feb 21 03:33:54 UTC 2003
5.0(1)
<defaultVersions>
<defaultVersion aspect="S">
<version>149.0</version>
<date>2005-03-04</date>
</defaultVersion>
</defaultVersions>
1.1 - 5.0(1)S149
Linux version 2.4.26-IDS-smp-bigphys (csailer@mcq) (gcc version 2.96 20000731 (R
ed Hat Linux 7.3 2.96-112)) #2 SMP Fri Mar 4 04:11:31 CST 2005
 03:33:54 up 21 days, 23:15,  3 users,  load average: 0.96, 0.86, 0.78
--MORE--
```

Version Information

The **show version** command is useful for establishing the general health of the sensor.

This section describes the **show version** command, and contains the following topics:

- Overview, page C-53
- Displaying Version Information, page C-53

Overview

The **show version** command shows the general health of the sensor and can indicate where a failure is occurring. It gives the following information:

- Which applications are running
- Versions of the applications
- Disk and memory usage
- Upgrade history of the applications

Note Choose **Monitoring > Support Information > Diagnostics Report** to get the same information from IDM or ASDM.

Displaying Version Information

Use the **show version** command to display version information for all installed operating system packages, signature packages, and IPS processes running on the system. To view the configuration for the entire system, use the **more current-config** command.

To display the version and configuration, follow these steps:

Step 1 Log in to the CLI.

Step 2 View version information:

```
sensor# show version
```

The following examples show sample version output for the appliance and the NM-CIDS.

Sample version output for the appliance:

```
sensor# show version
Application Partition:

Cisco Intrusion Prevention System, Version 5.0(0.29)S135.0

OS Version 2.4.26-IDS-smp-bigphys
Platform: IPS-4255-K9
Serial Number: JAB0815R017
No license present
Sensor up-time is 5 days.
Using 722145280 out of 3974291456 bytes of available memory (18% usage)
system is using 17.3M out of 29.0M bytes of available disk space (59% usage)
application-data is using 36.3M out of 166.8M bytes of available disk space (23% usage)
boot is using 39.4M out of 68.6M bytes of available disk space (61% usage)


MainApp          2005_Feb_18_03.00   (Release)   2005-02-18T03:13:47-0600   Running
AnalysisEngine   2005_Feb_18_03.00   (Release)   2005-02-18T03:13:47-0600   Running
CLI              2005_Feb_18_03.00   (Release)   2005-02-18T03:13:47-0600


Upgrade History:

  IDS-K9-maj-5.0-0.29-S91-0.29-.pkg   03:00:00 UTC Mon Feb 16 2004

Recovery Partition Version 1.1 - 5.0(0.29)S91(0.29)
```

```
sensor#
```

Sample version output for NM-CIDS:

```
nm-cids# show version
Application Partition:
Cisco Intrusion Prevention System, Version 5.0(0.27)S129.0

OS Version 2.4.26-IDS-smp-bigphys
Platform: NM-CIDS
Serial Number: JAD06490681
No license present
Sensor up-time is 1 day.
Using 485675008 out of 509448192 bytes of available memory (95% usage)
system is using 17.3M out of 29.0M bytes of available disk space (59% usage)
application-data is using 31.1M out of 166.8M bytes of available disk space (20% usage)
boot is using 39.5M out of 68.6M bytes of available disk space (61% usage)
application-log is using 529.6M out of 2.8G bytes of available disk space (20% usage)


MainApp          2005_Feb_09_03.00   (Release)   2005-02-09T03:22:27-0600   Running
AnalysisEngine   2005_Feb_09_03.00   (Release)   2005-02-09T03:22:27-0600   Running
CLI              2005_Feb_09_03.00   (Release)   2005-02-09T03:22:27-0600


Upgrade History:

  IDS-K9-maj-5.0-0.27-S91-0.27-.pkg   03:00:00 UTC Thu Feb 05 2004

Recovery Partition Version 1.1 - 5.0(0.27)S91(0.27)

nm-cids#
```

Note If the --MORE-- prompt is displayed, press the spacebar to see more information or **Ctrl-C** to cancel the output and get back to the CLI prompt.

Step 3 View configuration information:

Note You can use the **more current-config** or **show configuration** commands.

```
sensor# more current-config
! ------------------------------
! Version 5.0(0.26)
! Current configuration last modified Wed Feb 16 03:20:54 2005
! ------------------------------
display-serial
! ------------------------------
service analysis-engine
exit
! ------------------------------
service authentication
exit
! ------------------------------
service event-action-rules rules0
exit
! ------------------------------
service host
network-settings
host-ip 10.89.147.31/25,10.89.147.126
```

```
host-name sensor
access-list 0.0.0.0/0
login-banner-text This message will be displayed on banner login.
exit
time-zone-settings
--MORE--
```

Statistics Information

The **show statistics** command is useful for examining the state of the sensor services. This section describes the **show statistics** command, and contains the following topics:

- Overview, page C-55
- Displaying Statistics, page C-55

Overview

The **show statistics** command provides a snapshot of the state of the sensor's services. The following services provide statistics:

- AnalysisEngine
- Authentication
- Denied Attackers
- Event Server
- Event Store
- Host
- Logger
- Attack Response (formerly known as Network Access)
- Notification
- SDEE Server
- Transaction Server
- Transaction Source
- Virtual Sensor
- Web Server

Note Click **Monitoring > Support Information > Statistics** to get the same information from IDM.

Displaying Statistics

Use the **show statistics virtual-sensor** [**clear**] command to display the statistics for the virtual sensor. Use the **show statistics [analysis-engine | authentication | denied-attackers | event-server | event-store | host | logger | network-access | notification | sdee-server | transaction-server | transaction-source | web-server**] [**clear**] command to generate statistics for each sensor application.

Note The **clear** option is not available for the analysis engine, host, or network access applications.

To display statistics for the sensor, follow these steps:

Step 1 Log in to the CLI.

Step 2 Display the statistics for the virtual sensor:

```
sensor# show statistics virtual-sensor
Virtual Sensor Statistics
   Statistics for Virtual Sensor vs0
      Name of current Signature-Definition instance = sig0
      Name of current Event-Action-Rules instance = rules0
      List of interfaces monitored by this virtual sensor = fe0_1
      General Statistics for this Virtual Sensor
         Number of seconds since a reset of the statistics = 1675
         Measure of the level of resource utilization = 0
         Total packets processed since reset = 241
         Total IP packets processed since reset = 12
         Total packets that were not IP processed since reset = 229
         Total TCP packets processed since reset = 0
         Total UDP packets processed since reset = 0
         Total ICMP packets processed since reset = 12
         Total packets that were not TCP, UDP, or ICMP processed since reset = 0
         Total ARP packets processed since reset = 0
         Total ISL encapsulated packets processed since reset = 0
         Total 802.1q encapsulated packets processed since reset = 0
         Total packets with bad IP checksums processed since reset = 0
         Total packets with bad layer 4 checksums processed since reset = 0
         Total number of bytes processed since reset = 22513
         The rate of packets per second since reset = 0
         The rate of bytes per second since reset = 13
         The average bytes per packet since reset = 93
      Denied Address Information
         Number of Active Denied Attackers = 0
         Number of Denied Attackers Inserted = 0
         Number of Denied Attackers Total Hits = 0
         Number of times max-denied-attackers limited creation of new entry = 0
         Number of exec Clear commands during uptime = 0
      Denied Attackers and hit count for each.
      The Signature Database Statistics.
         The Number of each type of node active in the system (can not be reset)
            Total nodes active = 0
            TCP nodes keyed on both IP addresses and both ports = 0
            UDP nodes keyed on both IP addresses and both ports = 0
            IP nodes keyed on both IP addresses = 0
         The number of each type of node inserted since reset
            Total nodes inserted = 28
            TCP nodes keyed on both IP addresses and both ports = 0
            UDP nodes keyed on both IP addresses and both ports = 0
            IP nodes keyed on both IP addresses = 6
         The rate of nodes per second for each time since reset
            Nodes per second = 0
            TCP nodes keyed on both IP addresses and both ports per second = 0
            UDP nodes keyed on both IP addresses and both ports per second = 0
            IP nodes keyed on both IP addresses per second = 0
         The number of root nodes forced to expire because of memory constraints
            TCP nodes keyed on both IP addresses and both ports = 0
      Fragment Reassembly Unit Statistics for this Virtual Sensor
         Number of fragments currently in FRU = 0
         Number of datagrams currently in FRU = 0
```

```
   Number of fragments received since reset = 0
   Number of fragments forwarded since reset = 0
   Number of fragments dropped since last reset = 0
   Number of fragments modified since last reset = 0
   Number of complete datagrams reassembled since last reset = 0
   Fragments hitting too many fragments condition since last reset = 0
   Number of overlapping fragments since last reset = 0
   Number of Datagrams too big since last reset = 0
   Number of overwriting fragments since last reset = 0
   Number of Initial fragment missing since last reset = 0
   Fragments hitting the max partial dgrams limit since last reset = 0
   Fragments too small since last reset = 0
   Too many fragments per dgram limit since last reset = 0
   Number of datagram reassembly timeout since last reset = 0
   Too many fragments claiming to be the last since last reset = 0
   Fragments with bad fragment flags since last reset = 0
TCP Normalizer stage statistics
   Packets Input = 0
   Packets Modified = 0
   Dropped packets from queue = 0
   Dropped packets due to deny-connection = 0
   Current Streams = 0
   Current Streams Closed = 0
   Current Streams Closing = 0
   Current Streams Embryonic = 0
   Current Streams Established = 0
   Current Streams Denied = 0
Statistics for the TCP Stream Reassembly Unit
   Current Statistics for the TCP Stream Reassembly Unit
      TCP streams currently in the embryonic state = 0
      TCP streams currently in the established state = 0
      TCP streams currently in the closing state = 0
      TCP streams currently in the system = 0
      TCP Packets currently queued for reassembly = 0
   Cumulative Statistics for the TCP Stream Reassembly Unit since reset
      TCP streams that have been tracked since last reset = 0
      TCP streams that had a gap in the sequence jumped = 0
      TCP streams that was abandoned due to a gap in the sequence = 0
      TCP packets that arrived out of sequence order for their stream = 0
      TCP packets that arrived out of state order for their stream = 0
      The rate of TCP connections tracked per second since reset = 0
SigEvent Preliminary Stage Statistics
   Number of Alerts received = 491
   Number of Alerts Consumed by AlertInterval = 0
   Number of Alerts Consumed by Event Count = 0
   Number of FireOnce First Alerts = 6
   Number of FireOnce Intermediate Alerts = 480
   Number of Summary First Alerts = 0
   Number of Summary Intermediate Alerts = 0
   Number of Regular Summary Final Alerts = 0
   Number of Global Summary Final Alerts = 0
   Number of Alerts Output for further processing = 491
SigEvent Action Override Stage Statistics
   Number of Alerts received to Action Override Processor = 0
   Number of Alerts where an override was applied = 0
   Actions Added
      deny-attacker-inline = 0
      deny-connection-inline = 0
      deny-packet-inline = 0
      modify-packet-inline = 0
      log-attacker-packets = 0
      log-pair-packets = 0
      log-victim-packets = 0
      produce-alert = 0
```

```
            produce-verbose-alert = 0
            request-block-connection = 0
            request-block-host = 0
            request-snmp-trap = 0
            reset-tcp-connection = 0
      SigEvent Action Filter Stage Statistics
         Number of Alerts received to Action Filter Processor = 0
         Number of Alerts where an action was filtered = 0
         Number of Filter Line matches = 0
         Actions Filtered
            deny-attacker-inline = 0
            deny-connection-inline = 0
            deny-packet-inline = 0
            modify-packet-inline = 0
            log-attacker-packets = 0
            log-pair-packets = 0
            log-victim-packets = 0
            produce-alert = 0
            produce-verbose-alert = 0
            request-block-connection = 0
            request-block-host = 0
            request-snmp-trap = 0
            reset-tcp-connection = 0
      SigEvent Action Handling Stage Statistics.
         Number of Alerts received to Action Handling Processor = 491
         Number of Alerts where produceAlert was forced = 0
         Number of Alerts where produceAlert was off = 0
         Actions Performed
            deny-attacker-inline = 0
            deny-connection-inline = 0
            deny-packet-inline = 0
            modify-packet-inline = 0
            log-attacker-packets = 0
            log-pair-packets = 0
            log-victim-packets = 0
            produce-alert = 11
            produce-verbose-alert = 0
            request-block-connection = 0
            request-block-host = 5
            request-snmp-trap = 0
            reset-tcp-connection = 0
         Deny Actions Requested in Promiscuous Mode
            deny-packet not performed = 0
            deny-connection not performed = 0
            deny-attacker not performed = 0
            modify-packet not performed = 0
         Number of Alerts where deny-connection was forced for deny-packet action = 0
         Number of Alerts where deny-packet was forced for non-TCP deny-connection action
= 0
      Per-Signature SigEvent count since reset
         Sig 2004 = 5
         Sig 2156 = 486
sensor#
```

Step 3 Display the statistics for AnalysisEngine:

```
sensor# show statistics analysis-engine
Analysis Engine Statistics
   Number of seconds since service started = 1999
   Measure of the level of current resource utilization = 0
   Measure of the level of maximum resource utilization = 0
   The rate of TCP connections tracked per second = 0
   The rate of packets per second = 0
   The rate of bytes per second = 13
   Receiver Statistics
      Total number of packets processed since reset = 290
      Total number of IP packets processed since reset = 12
   Transmitter Statistics
      Total number of packets transmitted = 290
      Total number of packets denied = 0
      Total number of packets reset = 0
   Fragment Reassembly Unit Statistics
      Number of fragments currently in FRU = 0
      Number of datagrams currently in FRU = 0
   TCP Stream Reassembly Unit Statistics
      TCP streams currently in the embryonic state = 0
      TCP streams currently in the established state = 0
      TCP streams currently in the closing state = 0
      TCP streams currently in the system = 0
      TCP Packets currently queued for reassembly = 0
   The Signature Database Statistics.
      Total nodes active = 0
      TCP nodes keyed on both IP addresses and both ports = 0
      UDP nodes keyed on both IP addresses and both ports = 0
      IP nodes keyed on both IP addresses = 0
   Statistics for Signature Events
      Number of SigEvents since reset = 491
   Statistics for Actions executed on a SigEvent
      Number of Alerts written to the IdsEventStore = 11
sensor#
```

Step 4 Display the statistics for authentication:

```
sensor# show statistics authentication
General
   totalAuthenticationAttempts = 2
   failedAuthenticationAttempts = 0
sensor#
```

Step 5 Display the statistics for the denied attackers in the system:

```
sensor# show statistics denied-attackers
Denied Attackers and hit count for each.
sensor#
```

Step 6 Display the statistics for the event server:

```
sensor# show statistics event-server
General
   openSubscriptions = 0
   blockedSubscriptions = 0
Subscriptions
sensor#
```

Step 7 Display the statistics for Event Store:

```
sensor# show statistics event-store
Event store statistics
   General information about the event store
```

```
      The current number of open subscriptions = 2
      The number of events lost by subscriptions and queries = 0
      The number of queries issued = 0
      The number of times the event store circular buffer has wrapped = 0
   Number of events of each type currently stored
      Debug events = 0
      Status events = 9904
      Log transaction events = 0
      Shun request events = 61
      Error events, warning = 67
      Error events, error = 83
      Error events, fatal = 0
      Alert events, informational = 60
      Alert events, low = 1
      Alert events, medium = 60
      Alert events, high = 0
sensor#
```

Step 8 Display the statistics for the host:

```
sensor# show statistics host
General Statistics
   Last Change To Host Config (UTC) = 16:11:05 Thu Feb 10 2005
   Command Control Port Device = FastEthernet0/0
Network Statistics
   fe0_0     Link encap:Ethernet HWaddr 00:0B:46:53:06:AA
             inet addr:10.89.149.185 Bcast:10.89.149.255 Mask:255.255.255.128
             UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
             RX packets:1001522 errors:0 dropped:0 overruns:0 frame:0
             TX packets:469569 errors:0 dropped:0 overruns:0 carrier:0
             collisions:0 txqueuelen:1000
             RX bytes:57547021 (54.8 MiB) TX bytes:63832557 (60.8 MiB)
             Interrupt:9 Base address:0xf400 Memory:c0000000-c0000038
NTP Statistics
   status = Not applicable
Memory Usage
   usedBytes = 500592640
   freeBytes = 8855552
   totalBytes = 509448192
Swap Usage
   Used Bytes = 77824
   Free Bytes = 600649728

   Total Bytes = 600727552
CPU Statistics
   Usage over last 5 seconds = 0
   Usage over last minute = 1
   Usage over last 5 minutes = 1
Memory Statistics
   Memory usage (bytes) = 500498432
   Memory free (bytes) = 894976032
Auto Update Statistics
   lastDirectoryReadAttempt = N/A
   lastDownloadAttempt = N/A
   lastInstallAttempt = N/A
   nextAttempt = N/A
sensor#
```

Step 9 Display the statistics for the logging application:

```
sensor# show statistics logger
The number of Log interprocessor FIFO overruns = 0
The number of syslog messages received = 11
The number of <evError> events written to the event store by severity
```

```
   Fatal Severity = 0
   Error Severity = 64
   Warning Severity = 35
   TOTAL = 99
The number of log messages written to the message log by severity
   Fatal Severity = 0
   Error Severity = 64
   Warning Severity = 24
   Timing Severity = 311
   Debug Severity = 31522
   Unknown Severity = 7
   TOTAL = 31928
sensor#
```

Step 10 Display the statistics for ARC:

```
sensor# show statistics network-access
Current Configuration
   LogAllBlockEventsAndSensors = true
   EnableNvramWrite = false
   EnableAclLogging = false
   AllowSensorBlock = false
   BlockMaxEntries = 11
   MaxDeviceInterfaces = 250
   NetDevice
      Type = PIX
      IP = 10.89.150.171
      NATAddr = 0.0.0.0
      Communications = ssh-3des
   NetDevice
      Type = PIX
      IP = 10.89.150.219
      NATAddr = 0.0.0.0
      Communications = ssh-des
   NetDevice
      Type = PIX
      IP = 10.89.150.250
      NATAddr = 0.0.0.0
      Communications = telnet
   NetDevice
      Type = Cisco
      IP = 10.89.150.158
      NATAddr = 0.0.0.0
      Communications = telnet
      BlockInterface
         InterfaceName = ethernet0/1
         InterfaceDirection = out
         InterfacePostBlock = Post_Acl_Test
      BlockInterface
         InterfaceName = ethernet0/1
         InterfaceDirection = in
         InterfacePreBlock = Pre_Acl_Test
         InterfacePostBlock = Post_Acl_Test
   NetDevice
      Type = CAT6000_VACL
      IP = 10.89.150.138
      NATAddr = 0.0.0.0
      Communications = telnet
      BlockInterface
         InterfaceName = 502
         InterfacePreBlock = Pre_Acl_Test
      BlockInterface
         InterfaceName = 507
         InterfacePostBlock = Post_Acl_Test
```

```
State
   BlockEnable = true
   NetDevice
      IP = 10.89.150.171
      AclSupport = Does not use ACLs
      Version = 6.3
      State = Active
      Firewall-type = PIX
   NetDevice
      IP = 10.89.150.219
      AclSupport = Does not use ACLs
      Version = 7.0
      State = Active
      Firewall-type = ASA
   NetDevice
      IP = 10.89.150.250
      AclSupport = Does not use ACLs
      Version = 2.2
      State = Active
      Firewall-type = FWSM
   NetDevice
      IP = 10.89.150.158
      AclSupport = uses Named ACLs
      Version = 12.2
      State = Active
   NetDevice
      IP = 10.89.150.138
      AclSupport = Uses VACLs
      Version = 8.4
      State = Active
   BlockedAddr
      Host
         IP = 22.33.4.5
         Vlan =
         ActualIp =
         BlockMinutes =
      Host
         IP = 21.21.12.12
         Vlan =
         ActualIp =
         BlockMinutes =
      Host
         IP = 122.122.33.4
         Vlan =
         ActualIp =
         BlockMinutes = 60
         MinutesRemaining = 24
      Network
         IP = 111.22.0.0
         Mask = 255.255.0.0
         BlockMinutes =
sensor#
```

Step 11 Display the statistics for the notification application:

```
sensor# show statistics notification
General
   Number of SNMP set requests = 0
   Number of SNMP get requests = 0
   Number of error traps sent = 0
   Number of alert traps sent = 0
sensor#
```

Step 12 Display the statistics for the SDEE server:

```
sensor# show statistics sdee-server
General
   Open Subscriptions = 0
   Blocked Subscriptions = 0
   Maximum Available Subscriptions = 5
   Maximum Events Per Retrieval = 500
Subscriptions
sensor#
```

Step 13 Display the statistics for the transaction server:

```
sensor# show statistics transaction-server
General
   totalControlTransactions = 35
   failedControlTransactions = 0
sensor#
```

Step 14 Display the statistics for the transaction source:

```
sensor# show statistics transaction-source
General
   totalControlTransactions = 0
   failedControlTransactions = 0
sensor#
```

Step 15 Display the statistics for Web Server:

```
sensor# show statistics web-server
listener-443
   number of server session requests handled = 61
   number of server session requests rejected = 0
   total HTTP requests handled = 35
   maximum number of session objects allowed = 40
   number of idle allocated session objects = 10
   number of busy allocated session objects = 0
crypto library version = 6.0.3
sensor#
```

Step 16 To clear the statistics for an application, for example, the logging application:

```
sensor# show statistics logger clear
The number of Log interprocessor FIFO overruns = 0
The number of syslog messages received = 141
The number of <evError> events written to the event store by severity
   Fatal Severity = 0
   Error Severity = 14
   Warning Severity = 142
   TOTAL = 156
The number of log messages written to the message log by severity
   Fatal Severity = 0
   Error Severity = 14
   Warning Severity = 1
   Timing Severity = 0
   Debug Severity = 0
   Unknown Severity = 28
   TOTAL = 43
```

The statistics were retrieved and cleared.

Step 17 Verify that the statistics have been cleared:

```
sensor# show statistics logger
The number of Log interprocessor FIFO overruns = 0
The number of syslog messages received = 0
```

```
The number of <evError> events written to the event store by severity
   Fatal Severity = 0
   Error Severity = 0
   Warning Severity = 0
   TOTAL = 0
The number of log messages written to the message log by severity
   Fatal Severity = 0
   Error Severity = 0
   Warning Severity = 0
   Timing Severity = 0
   Debug Severity = 0
   Unknown Severity = 0
   TOTAL = 0
sensor#
```

The statistics all begin from 0.

Interfaces Information

The **show interfaces** command is useful for gathering information on the sensing and command and control interfaces.

This section describes the **show interfaces** command, and contains the following topics:

Overview

You can learn the following information from the **show interfaces** command:

- Whether the interface is up or down
- Whether or not packets are being seen, and on which interfaces
- Whether or not packets are being dropped by SensorApp
- Whether or not there are errors being reported by the interfaces that can result in packet drops

The **show interfaces** command displays statistics for all system interfaces. Or you can use the individual commands to display statistics for the command and control interface (**show interfaces** *command_control_interface_name*), the sensing interface (**show interfaces** *interface_name*).

Interfaces Command Output

The following example shows the output from the **show interfaces** command:

```
sensor# show interfaces
Interface Statistics
   Total Packets Received = 0
   Total Bytes Received = 0
   Missed Packet Percentage = 0
   Current Bypass Mode = Auto_off
MAC statistics from interface GigabitEthernet0/1
   Media Type = backplane
   Missed Packet Percentage = 0
   Inline Mode = Unpaired
   Pair Status = N/A
```

```
   Link Status = Up
   Link Speed = Auto_1000
   Link Duplex = Auto_Full
   Total Packets Received = 0
   Total Bytes Received = 0
   Total Multicast Packets Received = 0
   Total Broadcast Packets Received = 0
   Total Jumbo Packets Received = 0
   Total Undersize Packets Received = 0
   Total Receive Errors = 0
   Total Receive FIFO Overruns = 0
   Total Packets Transmitted = 0
   Total Bytes Transmitted = 0
   Total Multicast Packets Transmitted = 0
   Total Broadcast Packets Transmitted = 0
   Total Jumbo Packets Transmitted = 0
   Total Undersize Packets Transmitted = 0
   Total Transmit Errors = 0
   Total Transmit FIFO Overruns = 0
MAC statistics from interface GigabitEthernet0/0
   Media Type = TX
   Link Status = Up
   Link Speed = Auto_100
   Link Duplex = Auto_Full
   Total Packets Received = 2211296
   Total Bytes Received = 157577635
   Total Multicast Packets Received = 20
   Total Receive Errors = 0
   Total Receive FIFO Overruns = 0
   Total Packets Transmitted = 239723
   Total Bytes Transmitted = 107213390
   Total Transmit Errors = 0
   Total Transmit FIFO Overruns = 0
sensor#
```

Events Information

You can use the **show events** command to view the alerts generated by SensorApp and errors generated by an application.

This section describes the **show events** command, and contains the following topics:

Sensor Events

There are five types of events:

- evAlert—Intrusion detection alerts
- evError—Application errors
- evStatus—Status changes, such as an IP log being created
- evLogTransaction—Record of control transactions processed by each sensor application
- evShunRqst—Block requests

Events remain in the Event Store until they are overwritten by newer events.

Overview

The **show events** command is useful for troubleshooting event capture issues in which you are not seeing events in Event Viewer or Security Monitor. You can use the **show events** command to determine which events are being generated on the sensor to make sure events are being generated and that the fault lies with the monitoring side.

You can clear all events from Event Store by using the **clear events** command.

Here are the parameters for the **show events** command:

```
sensor# show events
<cr>
alert          Display local system alerts.
error          Display error events.
hh:mm[:ss]     Display start time.
log            Display log events.
nac            Display NAC shun events.
past           Display events starting in the past specified time.
status         Display status events.
|              Output modifiers.
```

Displaying Events

Use the **show events** [{[**alert** [informational] [low] [medium] [high] [**include-traits** ***traits***] [**exclude-traits** ***traits***]] | **error** [warning] [error] [fatal] | **log** | **NAC** | **status**}] [*hh:mm:ss* [*month day* [*year*]] | **past** *hh:mm:ss*] command to display events from the Event Store.

Events are displayed beginning at the start time. If you do not specify a start time, events are displayed beginning at the current time. If you do not specify an event type, all events are displayed.

Note Events are displayed as a live feed until you cancel the request by pressing Ctrl-C.

The following options apply:

- **alert**—Displays alerts. Provides notification of some suspicious activity that may indicate an attack is in process or has been attempted.

  If no level is selected (informational, low, medium, or high), all alert events are displayed.
- **include-traits**—Displays alerts that have the specified traits.
- **exclude-traits**—Does not display alerts that have the specified traits.
- **traits**—Trait bit position in decimal (0 to 15).
- **error**—Displays error events. Error events are generated by services when error conditions are encountered.
- **log**—Displays log events. Log events are generated when a transaction is received and responded to by an application. Contains information about the request, response, and success or failure of the transaction.
- **NAC**—Displays Attack Response Controller (ARC) requests.

Note ARC is formerly known as Network Access Controller (NAC). This name change has not been completely implemented throughout the IDM and CLI for IPS 5.1.

- **status**—Displays status events.
- **past**—Displays events starting in the past for the specified hours, minutes, and seconds.
- *hh:mm:ss*—Hours, minutes, and seconds in the past to begin the display.

Note The **show events** command waits until a specified event is available. It continues to wait and display events until you exit by pressing Ctrl-C.

To display events from the Event Store, follow these steps:

Step 1 Log in to the CLI.

Step 2 Display all events starting now:

```
sensor#@ show events
evError: eventId=1041472274774840147 severity=warning vendor=Cisco
  originator:
    hostId: sensor2
    appName: cidwebserver
    appInstanceId: 12075
  time: 2003/01/07 04:41:45 2003/01/07 04:41:45 UTC
  errorMessage: name=errWarning received fatal alert: certificate_unknown

evError: eventId=1041472274774840148 severity=error vendor=Cisco
  originator:
    hostId: sensor2
    appName: cidwebserver
    appInstanceId: 351
  time: 2003/01/07 04:41:45 2003/01/07 04:41:45 UTC
  errorMessage: name=errTransport WebSession::sessionTask(6) TLS connection exce
ption: handshake incomplete.
```

The feed continues showing all events until you press **Ctrl-C**.

Step 3 Display the block requests beginning at 10:00 a.m. on February 9, 2005:

```
sensor#@ show events NAC 10:00:00 Feb 9 2005
evShunRqst: eventId=1106837332219222281 vendor=Cisco
  originator:
    deviceName: Sensor1
    appName: NetworkAccessControllerApp
    appInstance: 654
  time: 2005/02/09 10:33:31 2004/08/09 13:13:31
  shunInfo:
    host: connectionShun=false
      srcAddr: 11.0.0.1
      destAddr:
      srcPort:
      destPort:
      protocol: numericType=0 other
    timeoutMinutes: 40
  evAlertRef: hostId=esendHost 123456789012345678
sensor#
```

Step 4 Display errors with the warning level starting at 10:00 a.m. February 9 2005:

```
sensor# show events error warning 10:00:00 Feb 9 2005
evError: eventId=1041472274774840197 severity=warning vendor=Cisco
  originator:
    hostId: sensor
    appName: cidwebserver
    appInstanceId: 12160
  time: 2003/01/07 04:49:25 2003/01/07 04:49:25 UTC
  errorMessage: name=errWarning received fatal alert: certificate_unknown
```

Step 5 Display alerts from the past 45 seconds:

```
sensor# show events alert past 00:00:45

evIdsAlert: eventId=1109695939102805307 severity=medium vendor=Cisco
  originator:
    hostId: sensor
    appName: sensorApp
    appInstanceId: 367
  time: 2005/03/02 14:15:59 2005/03/02 14:15:59 UTC
  signature: description=Nachi Worm ICMP Echo Request id=2156 version=S54
    subsigId: 0
    sigDetails: Nachi ICMP
  interfaceGroup:
  vlan: 0
  participants:
    attacker:
      addr: locality=OUT 10.89.228.202
    target:
      addr: locality=OUT 10.89.150.185
  riskRatingValue: 70
  interface: fe0_1
  protocol: icmp


evIdsAlert: eventId=1109695939102805308 severity=medium vendor=Cisco
  originator:
--MORE--
```

Step 6 Display events that began 30 seconds in the past:

```
sensor# show events past 00:00:30
evStatus: eventId=1041526834774829055 vendor=Cisco
  originator:
    hostId: sensor
    appName: mainApp
    appInstanceId: 2215
  time: 2003/01/08 02:41:00 2003/01/08 02:41:00 UTC
  controlTransaction: command=getVersion successful=true
    description: Control transaction response.
    requestor:
      user: cids
      application:
        hostId: 64.101.182.101
        appName: -cidcli
        appInstanceId: 2316


evStatus: eventId=1041526834774829056 vendor=Cisco
  originator:
    hostId: sensor
    appName: login(pam_unix)
    appInstanceId: 2315
  time: 2003/01/08 02:41:00 2003/01/08 02:41:00 UTC
```

```
  syslogMessage:
    description: session opened for user cisco by cisco(uid=0)
```

Clearing Events

Use the **clear events** command to clear Event Store.

To clear events from Event Store, follow these steps:

Step 1 Log in to the CLI using an account with administrator privileges.

Step 2 Clear Event Store:

```
sensor# clear events
Warning: Executing this command will remove all events currently stored in the event
store.
Continue with clear? []:
```

Step 3 Type **yes** to clear the events.

cidDump Script

If you do not have access to IDM or the CLI, you can run the underlying script cidDump from the Service account by logging in as root and running /usr/cids/idsRoot/bin/cidDump. The cidDump file path is /usr/cids/idsRoot/htdocs/private/cidDump.html.

cidDump is a script that captures a large amount of information including the IPS processes list, log files, OS information, directory listings, package information, and configuration files.

To run the cidDump script, follow these steps:

Step 1 Log in to the sensor Service account.

Step 2 **Su** to **root** using the Service account password.

Step 3 Enter the following command:

```
/usr/cids/idsRoot/bin/cidDump
```

Step 4 Enter the following command to compress the resulting /usr/cids/idsRoot/log/cidDump.html file:

```
gzip /usr/cids/idsRoot/log/cidDump.html
```

Step 5 Send the resulting HTML file to TAC or the IPS developers in case of a problem.

For the procedure, see Uploading and Accessing Files on the Cisco FTP Site, page C-70.

Uploading and Accessing Files on the Cisco FTP Site

You can upload large files, for example, cidDump.html, the show tech-support command output, and cores, to the ftp-sj server.

To upload and access files on the Cisco FTP site, follow these steps:

Step 1 Log in to ftp-sj.cisco.com as anonymous.

Step 2 Change to the /incoming directory.

Step 3 Use the **put** command to upload the files.

Make sure to use the binary transfer type.

Step 4 To access uploaded files, log in to an ECS-supported host.

Step 5 Change to the /auto/ftp/incoming directory.

GLOSSARY

Numerals

3DES Triple Data Encryption Standard. A stronger version of DES, which is the default encryption method for SSH version 1.5. Used when establishing an SSH session with the sensor. It can be used when the sensor is managing a device.

A

aaa authentication, authorization, and accounting. The primary and recommended method for access control in Cisco devices.

AAA authentication, authorization, and accounting. Pronounced “triple a.”

ACE Access Control Entry. An entry in the ACL that describes what action should be taken for a specified address or protocol. The sensor adds/removes ACE to block hosts.

ACK acknowledgement. Notification sent from one network device to another to acknowledge that some event occurred (for example, the receipt of a message).

ACL Access Control List. A list of ACEs that control the flow of data through a router. There are two ACLs per router interface for inbound data and outbound data. Only one ACL per direction can be active at a time. ACLs are identified by number or by name. ACLs can be standard, enhanced, or extended. You can configure the sensor to manage ACLs.

action The sensor’s response to an event. An action only happens if the event is not filtered. Examples include TCP reset, block host, block connection, IP logging, and capturing the alert trigger packet.

active ACL The ACL created and maintained by ARC and applied to the router block interfaces.

AIC engine Application Inspection and Control engine. Provides deep analysis of web traffic. It provides granular control over HTTP sessions to prevent abuse of the HTTP protocol. It allows administrative control over applications that try to tunnel over specified ports, such as instant messaging, and tunneling applications, such as gotomypc. It can also inspect FTP traffic and control the commands being issued.

AIP-SSM Advanced Inspection and Prevention Security Services Module. The IPS plug-in module in the Cisco ASA 5500 series adaptive security appliance. See ASA.

Alarm Channel The IPS software module that processes all signature events generated by the inspectors. Its primary function is to generate alerts for each event it receives.

alert Specifically, an IPS event type; it is written to the Event Store as an evidsAlert. In general, an alert is an IPS message that indicates a network exploit in progress or a potential security problem occurrence. Also known as an alarm.

Analysis Engine The IPS software module that handles sensor configuration. It maps the interfaces and also the signature and alarm channel policy to the configured interfaces. It performs packet analysis and alert detection.

API Application Programming Interface. The means by which an application program talks to communications software. Standardized APIs allow application programs to be developed independently of the underlying method of communication. Computer application programs run a set of standard software interrupts, calls, and data formats to initiate contact with other devices (for example, network services, mainframe communications programs, or other program-to-program communications). Typically, APIs make it easier for software developers to create links that an application needs to communicate with the operating system or with the network.

application Any program (process) designed to run in the Cisco IPS environment.

application instance A specific application running on a specific piece of hardware in the IPS environment. An application instance is addressable by its name and the IP address of its host computer.

ARC Attack Response Controller. Formerly known as Network Access Controller (NAC). A component of the IPS. A software module that provides block and unblock functionality where applicable.

architecture The overall structure of a computer or communication system. The architecture influences the capabilities and limitations of the system.

ARP Address Resolution Protocol. Internet protocol used to map an IP address to a MAC address. Defined in RFC 826.

ASA Adaptive Security Appliance. The ASA combines firewall, VPN concentrator, and intrusion prevention software functionality into one software image. You can configure ASA in single mode or multi-mode.

ASDM Adaptive Security Device Manager. A web-based application that lets you configure and manage your ASA.

atomic attack Represents exploits contained within a single packet. For example, the “ping of death” attack is a single, abnormally large ICMP packet.

Atomic engine There are two Atomic engines: Atomic IP inspects IP protocol packets and associated Layer-4 transport protocols, and Atomic ARP inspects Layer-2 ARP protocol.

attack An assault on system security that derives from an intelligent threat, that is, an intelligent act that is a deliberate attempt (especially in the sense of method or technique) to evade security services and violate the security policy of a system.

authentication Process of verifying that a user has permission to use the system, usually by means of a password key or certificate.

AuthenticationApp A component of the IPS. It verifies that users have the correct permissions to perform CLI, IDM, or RDEP actions.

B

backplane The physical connection between an interface processor or card and the data buses and the power distribution buses inside a chassis.

base version A software release that must be installed before a follow-up release such as a service pack or signature update can be installed. Major and minor version upgrades are base version releases.

benign trigger A situation in which a signature is fired correctly, but the source of the traffic is nonmalicious.

BIOS Basic Input/Output System. The program that starts the sensor and communicates between the devices in the sensor and the system.

block The ability of the sensor to direct a network device to deny entry to all packets from a specified network host or network.

block interface The interface on the network device that the sensor manages.

BO2K BackOrifice 2000. A windows back door Trojan that runs over TCP and UDP.

Bpdu Bridge Protocol Data Unit. Spanning-Tree Protocol hello packet that is sent out at configurable intervals to exchange information among bridges in the network.

bypass mode Mode that lets packets continue to flow through the sensor even if the sensor fails. Bypass mode is only applicable to inline-paired interfaces.

C

CA certification authority. Entity that issues digital certificates (especially X.509 certificates) and vouches for the binding between the data items in a certificate. Sensors use self-signed certificates.

CA certificate Certificate for one CA issued by another CA.

certificate Digital representation of user or device attributes, including a public key, that is signed with an authoritative private key.

cidDump A script that captures a large amount of information including the IPS processes list, log files, OS information, directory listings, package information, and configuration files.

CIDEE Cisco Intrusion Detection Event Exchange. Specifies the extensions to SDEE that are used by Cisco IPS systems. The CIDEE standard specifies all possible extensions that may be supported by Cisco IPS systems.

CIDS header The header that is attached to each packet in the IPS system. It contains packet classification, packet length, checksum results, timestamp, and the receive interface.

cipher key The secret binary data used to convert between clear text and cipher text. When the same cipher key is used for both encryption and decryption, it is called symmetric. When it is used for either encryption or decryption (but not both), it is called asymmetric.

Cisco IOS Cisco system software that provides common functionality, scalability, and security for all products under the CiscoFusion architecture. Cisco IOS allows centralized, integrated, and automated installation and management of internetworks while supporting a wide variety of protocols, media, services, and platforms.

CLI command-line interface. A shell provided with the sensor used for configuring and controlling the sensor applications.

command and control interface The interface on the sensor that communicates with the IPS manager and other network devices. This interface has an assigned IP address.

community In SNMP, a logical group of managed devices and NMSs in the same administrative domain.

composite attack Spans multiple packets in a single session. Examples include most conversation attacks such as FTP, Telnet, and most Regex-based attacks.

connection block ARC blocks traffic from a given source IP address to a given destination IP address and destination port.

console A terminal or laptop computer used to monitor and control the sensor.

console port An RJ45 or DB9 serial port on the sensor that is used to connect to a console device.

control interface When ARC opens a Telnet or SSH session with a network device, it uses one of the device's routing interfaces as the remote IP address. This is the control interface.

control transaction An IPS message containing a command addressed to a specific application instance. Example control transactions include *start*, *stop*, *getConfig*.

cookie A piece of information sent by a web server to a web browser that the browser is expected to save and send back to the web server whenever the browser makes additional requests of the web server.

D

Database Processor See DBP.

datagram Logical grouping of information sent as a network layer unit over a transmission medium without prior establishment of a virtual circuit. IP datagrams are the primary information units in the Internet. The terms cell, frame, message, packet, and segment also are used to describe logical information groupings at various layers of the OSI reference model and in various technology circles.

DBP Database Processor. Maintains the signature state and flow databases.

DCE data circuit-terminating equipment (ITU-T expansion). Devices and connections of a communications network that comprise the network end of the user-to-network interface. The DCE provides a physical connection to the network, forwards traffic, and provides a clocking signal used to synchronize data transmission between DCE and DTE devices. Modems and interface cards are examples of DCE.

DDoS Distributed Denial of Service. An attack in which a multitude of compromised systems attack a single target, thereby causing denial of service for users of the targeted system. The flood of incoming messages to the target system essentially forces it to shut down, thereby denying service to the system to legitimate users.

Deny Filters Processor See DFP.

DES Data Encryption Standard. A strong encryption method where the strength lies in a 56-bit key rather than an algorithm.

destination address Address of a network device that is receiving data.

DFP Deny Filters Processor. Handles the deny attacker functions. It maintains a list of denied source IP addresses.

DIMM Dual In-line Memory Modules.

DMZ demilitarized zone. A separate network located in the neutral zone between a private (inside) network and a public (outside) network.

DNS Domain Name System. An Internet-wide hostname to IP address mapping. DNS enables you to convert human-readable names into the IP addresses needed for network packets.

DoS Denial of Service. An attack whose goal is just to disrupt the operation of a specific system or network.

DRAM dynamic random-access memory. RAM that stores information in capacitors that must be refreshed periodically. Delays can occur because DRAMs are inaccessible to the processor when refreshing their contents. However, DRAMs are less complex and have greater capacity than SRAMs.

DTE Data Terminal Equipment. Refers to the role of a device on an RS-232C connection. A DTE writes data to the transmit line and reads data from the receive line.

E

egress Traffic leaving the network.

encryption Application of a specific algorithm to data to alter the appearance of the data making it incomprehensible to those who are not authorized to see the information.

engine A component of the sensor designed to support many signatures in a certain category. Each engine has parameters that can be used to create signatures or tune existing signatures.

enterprise network Large and diverse network connecting most major points in a company or other organization. Differs from a WAN in that it is privately owned and maintained.

escaped expression Used in regular expression. A character can be represented as its hexadecimal value, for example, \x61 equals ‘a,’ so \x61 is an escaped expression representing the character ‘a.’

ESD electrostatic discharge. Electrostatic discharge is the rapid movement of a charge from one object to another object, which produces several thousand volts of electrical charge that can cause severe damage to electronic components or entire circuit card assemblies.

event An IPS message that contains an alert, a block request, a status message, or an error message.

Event Server One of the components of the IPS.

Event Store One of the components of the IPS. A fixed-size, indexed store used to store IPS events.

evIdsAlert The XML entity written to the Event Store that represents an alert.

F

fail closed Blocks traffic on the device after a hardware failure.

fail open Lets traffic pass through the device after a hardware failure.

false negative A signature is not fired when offending traffic is detected.

false positive Normal traffic or a benign action causes a signature to fire.

Fast Ethernet Any of a number of 100-Mbps Ethernet specifications. Fast Ethernet offers a speed increase 10 times that of the 10BaseT Ethernet specification while preserving such qualities as frame format, MAC mechanisms, and MTU. Such similarities allow the use of existing 10BaseT applications and network management tools on Fast Ethernet networks. Based on an extension to the IEEE 802.3 specification.

firewall Router or access server, or several routers or access servers, designated as a buffer between any connected public networks and a private network. A firewall router uses access lists and other methods to ensure the security of the private network.

Flood engine Detects ICMP and UDP floods directed at hosts and networks.

flooding Traffic passing technique used by switches and bridges in which traffic received on an interface is sent out all the interfaces of that device except the interface on which the information was received originally.

fragment Piece of a larger packet that has been broken down to smaller units.

fragmentation Process of breaking a packet into smaller units when transmitting over a network medium that cannot support the original size of the packet.

Fragment Reassembly Processor See FRP.

FRP Fragment Reassembly Processor. Reassembles fragmented IP datagrams. It is also responsible for normalization of IP fragments when the sensor is in inline mode.

FTP File Transfer Protocol. Application protocol, part of the TCP/IP protocol stack, used for transferring files between network nodes. FTP is defined in RFC 959.

FTP server File Transfer Protocol server. A server that uses the FTP protocol for transferring files between network nodes.

full duplex Capability for simultaneous data transmission between a sending station and a receiving station.

FWSM Firewall Security Module. A module that can be installed in a Catalyst 6500 series switch. It uses the **shun** command to block. You can configure the FWSM in either single mode or multi-mode.

G

Gigabit Ethernet Standard for a high-speed Ethernet, approved by the IEEE (Institute of Electrical and Electronics Engineers) 802.3z standards committee in 1996.

GMT Greenwich Mean Time. Time zone at zero degrees longitude. Now called Coordinated Universal Time (UTC).

H

H.225.0 An ITU standard that governs H.225.0 session establishment and packetization. H.225.0 actually describes several different protocols: RAS, use of Q.931, and use of RTP.

H.245 An ITU standard that governs H.245 endpoint control.

H.323 Allows dissimilar communication devices to communicate with each other by using a standardized communication protocol. H.323 defines a common set of CODECs, call setup and negotiating procedures, and basic data transport methods.

half duplex Capability for data transmission in only one direction at a time between a sending station and a receiving station. BSC is an example of a half-duplex protocol.

handshake Sequence of messages exchanged between two or more network devices to ensure transmission synchronization.

hardware bypass Passes traffic at the network interface, does not pass it to the IPS system.

host block ARC blocks all traffic from a given IP address.

HTTP Hypertext Transfer Protocol. The stateless request/response media transfer protocol used in the IPS architecture for remote data exchange.

HTTPS An extension to the standard HTTP protocol that provides confidentiality by encrypting the traffic from the website. By default this protocol uses TCP port 443.

I

ICMP Internet Control Message Protocol. Network layer Internet protocol that reports errors and provides other information relevant to IP packet processing. Documented in RFC 792.

ICMP flood Denial of Service attack that sends a host more ICMP echo request (“ping”) packets than the protocol implementation can handle.

IDAPI Intrusion Detection Application Programming Interface. Provides a simple interface between IPS architecture applications. IDAPI reads and writes event data and provides a mechanism for control transactions.

IDCONF Intrusion Detection Configuration. A data format standard that defines operational messages that are used to configure intrusion detection and prevention systems.

IDIOM Intrusion Detection Interchange and Operations Messages. A data format standard that defines the event messages that are reported by intrusion detection systems and the operational messages that are used to configure and control intrusion detection systems.

IDM IPS Device Manager. A web-based application that lets you configure and manage your sensor. The web server for IDM resides on the sensor. You can access it through Netscape or Internet Explorer web browsers.

IDMEF Intrusion Detection Message Exchange Format. The IETF Intrusion Detection Working Group draft standard.

IDSM-2 Intrusion Detection System Module. A switching module that performs intrusion detection in the Catalyst 6500 series switch.

IDS MC Management Center for IDS Sensors. A web-based IDS manager that can manage configurations for up to 300 sensors.

inline mode All packets entering or leaving the network must pass through the sensor.

interface group Refers to the logical grouping of sensing interfaces. Multiple sensing interfaces can be assigned to a logical interface group. Signature parameters are tuned on a per-logical interface group basis.

intrusion detection system A security service that monitors and analyzes system events to find and provide real-time or near real-time warning of attempts to access system resources in an unauthorized manner.

IP address 32-bit address assigned to hosts using TCP/IP. An IP address belongs to one of five classes (A, B, C, D, or E) and is written as 4 octets separated by periods (dotted decimal format). Each address consists of a network number, an optional subnetwork number, and a host number. The network and subnetwork numbers together are used for routing, and the host number is used to address an individual host within the network or subnetwork. A subnet mask is used to extract network and subnetwork information from the IP address.

IPS Intrusion Prevention System. A system that alerts the user to the presence of an intrusion on the network through network traffic analysis techniques.

IPS data or message Describes the messages transferred over the command and control interface between IPS applications.

iplog A log of the binary packets to and from a designated address. Iplogs are created when the log Event Action is selected for a signature. Iplogs are stored in a libpcap format, which can be read by Wireshark and TCPDUMP.

IP spoofing IP spoofing attack occurs when an attacker outside your network pretends to be a trusted user either by using an IP address that is within the range of IP addresses for your network or by using an authorized external IP address that you trust and to which you want to provide access to specified resources on your network. Should an attacker get access to your IPSec security parameters, that attacker can masquerade as the remote user authorized to connect to the corporate network.

IPv6 IP version 6. Replacement for the current version of IP (version 4). IPv6 includes support for flow ID in the packet header, which can be used to identify flows. Formerly called IPng (next generation).

L

L2P Layer 2 Processor. Processes layer 2-related events. It also identifies malformed packets and removes them from the processing path.

LAN Local Area Network. Refers to the Layer 2 network domain local to a given host. Packets exchanged between two hosts on the same LAN do not require Layer 3 routing.

Layer 2 Processor See L2P.

Logger A component of the IPS.

logging Gathers actions that have occurred in a log file. Logging of security information is performed on two levels: logging of events (such as IPS commands, errors, and alerts), and logging of individual IP session information.

LOKI Remote access, back door Trojan, ICMP tunneling software. When the computer is infected, the malicious code creates an ICMP tunnel that can be used to send small payload ICMP replies

M

MainApp The main application in the IPS. The first application to start on the sensor after the operating system has booted.

maintenance partition image A full IPS image used to reimage the maintenance partition of the IDSM-2.

major update A base version that contains major new functionality or a major architectural change in the product.

manufacturing image Full IPS system image used by manufacturing to image sensors.

master blocking sensor A remote sensor that controls one or more devices. Blocking forwarding sensors send blocking requests to the master blocking sensor and the master blocking sensor executes the blocking requests.

MD5 Message Digest 5. A one-way hashing algorithm that produces a 128-bit hash. Both MD5 and Secure Hash Algorithm (SHA) are variations on MD4 and strengthen the security of the MD4 hashing algorithm. Cisco uses hashes for authentication within the IPSec framework. Also used for message authentication in SNMP v.2. MD5 verifies the integrity of the communication, authenticates the origin, and checks for timeliness.

MEG Mega Event Generator. Signature based on the Meta engine. The Meta engine takes alerts as input rather than packets.

Meta engine Defines events that occur in a related manner within a sliding time interval. This engine processes events rather than packets.

MIB Management Information Base. Database of network management information that is used and maintained by a network management protocol, such as SNMP or CMIP. The value of a MIB object can be changed or retrieved using SNMP or CMIP commands, usually through a GUI network management system. MIB objects are organized in a tree structure that includes public (standard) and private (proprietary) branches.

MIME Multipurpose Internet Mail Extension. Standard for transmitting nontext data (or data that cannot be represented in plain ASCII code) in Internet mail, such as binary, foreign language text (such as Russian or Chinese), audio, or video data. MIME is defined in RFC 2045.

minor update A minor version that contains minor enhancements to the product line. Minor updates are incremental to the major version, and are also base versions for service packs.

module A removable card in a switch, router, or security appliance chassis. AIP SSM, IDSM-2, and NM-CIDS are IPS modules.

monitoring interface See sensing interface.

MSFC, MSFC2 Multilayer Switch Feature Card. An optional card on a Catalyst 6000 supervisor engine that performs L3 routing for the switch.

MSRPC Microsoft Remote Procedure Call.

N

NAC Network Access Controller. See ARC.

NAT Native Address Translation. A network device can present an IP address to the outside networks that is different from the actual IP address of a host.

NBD Next Business Day. The arrival of replacement hardware according to Cisco service contracts.

network device A device that controls IP traffic on a network and can block an attacking host. An example of a network device is a Cisco router or PIX Firewall.

never block address Hosts and networks you have identified that should never be blocked.

never shun address See never block address.

NIC Network Interface Card. Board that provides network communication capabilities to and from a computer system.

NM-CIDS A network module that integrates IPS functionality into the branch office router.

NMS network management system. System responsible for managing at least part of a network. An NMS is generally a reasonably powerful and well-equipped computer, such as an engineering workstation. NMSs communicate with agents to help keep track of network statistics and resources.

node A physical communicating element on the command and control network. For example, an appliance, an IDSM-2, or a router.

Normalizer engine Configures how the IP and TCP normalizer functions and provides configuration for signature events related to the IP and TCP normalizer.

NTP Network Timing Protocol. Protocol built on top of TCP that ensures accurate local time-keeping with reference to radio and atomic clocks located on the Internet. This protocol is capable of synchronizing distributed clocks within milliseconds over long time periods.

NTP server Network Timing Protocol server. A server that uses NTP. NTP is a protocol built on top of TCP that ensures accurate local time-keeping with reference to radio and atomic clocks located on the Internet. This protocol is capable of synchronizing distributed clocks within milliseconds over long time periods.

NVRAM Non-Volatile Read/Write Memory. RAM that retains its contents when a unit is powered off.

O

OIR online insertion and removal. Feature that permits you to add, replace, or remove cards without interrupting the system power, entering console commands, or causing other software or interfaces to shutdown.

P

packet Logical grouping of information that includes a header containing control information and (usually) user data. Packets most often are used to refer to network layer units of data. The terms datagram, frame, message, and segment also are used to describe logical information groupings at various layers of the OSI reference model and in various technology circles.

PASC Port Spoof An attempt to open connections through a firewall to a protected FTP server to a non-FTP port. This happens when the firewall incorrectly interprets an FTP 227 (Entering Passive Mode) command by opening an unauthorized connection.

passive fingerprinting Act of determining the OS or services available on a system from passive observation of network interactions.

PAT Port Address Translation. A more restricted translation scheme than NAT in which a single IP address and different ports are used to represent the hosts of a network.

PCI Peripheral Component Interface. The most common peripheral expansion bus used on Intel-based computers.

PDU protocol data unit. OSI term for packet. See also BPDU and packet.

PEP Cisco Product Evolution Program. PEP is the UDI information that consists of the PID, the VID, and the SN of your sensor. PEP provides hardware version and serial number visibility through electronic query, product labels, and shipping items.

PER packed encoding rules. Instead of using a generic style of encoding that encodes all types in a uniform way, PER specializes the encoding based on the date type to generate much more compact representations.

PFC Policy Feature Card. An optional card on a Catalyst 6000 supervisor engine that supports VACL packet filtering.

PID Product Identifier. The orderable product identifier that is one of the three parts of the UDI. The UDI is part of the PEP policy.

ping packet internet groper. ICMP echo message and its reply. Often used in IP networks to test the reachability of a network device.

PIX Firewall Private Internet Exchange Firewall. A Cisco network security device that can be programmed to block/enable addresses and ports between networks.

PKI Public Key Infrastructure. Authentication of HTTP clients using the clients' X.509 certificates.

POST Power-On Self Test. Set of hardware diagnostics that runs on a hardware device when that device is powered up.

Post-ACL Designates an ACL from which ARC should read the ACL entries, and where it places entries after all deny entries for the addresses being blocked.

Pre-ACL Designates an ACL from which ARC should read the ACL entries, and where it places entries before any deny entries for the addresses being blocked.

promiscuous mode A passive interface for monitoring packets of the network segment. The sensing interface does not have an IP address assigned to it and is therefore invisible to attackers.

Q

Q.931 ITU-T specification for signaling to establish, maintain, and clear ISDN network connections.

R

rack mounting Refers to mounting a sensor in an equipment rack.

RAM random-access memory. Volatile memory that can be read and written by a microprocessor.

RAS Registration, Admission, and Status Protocol. Protocol that is used between endpoints and the gatekeeper to perform management functions. RAS signalling function performs registration, admissions, bandwidth changes, status, and disengage procedures between the VoIP gateway and the gatekeeper.

RDEP2 Remote Data Exchange Protocol version 2. The published specification for remote data exchange over the command and control network using HTTP and TLS.

reassembly The putting back together of an IP datagram at the destination after it has been fragmented either at the source or at an intermediate node.

recovery partition image An IPS image file that includes the full application image and installer used for recovery on sensors.

regex See regular expression.

regular expression A mechanism by which you can define how to search for a specified sequence of characters in a data stream or file. Regular expressions are a powerful and flexible notation almost like a mini-programming language that allow you to describe text. In the context of pattern matching, regular expressions allow a succinct description of any arbitrary pattern.

ROMMON Read-Only-Memory Monitor. ROMMON lets you TFTP system images onto the sensor for recovery purposes.

round-trip time See RTT.

RPC remote-procedure call. Technological foundation of client/server computing. RPCs are procedure calls that are built or specified by clients and are executed on servers, with the results returned over the network to the clients.

RR Risk Rating. An RR is a value between 0 and 100 that represents a numerical quantification of the risk associated with a particular event on the network.

RSM Router Switch Module. A router module that is installed in a Catalyst 5000 switch. It functions exactly like a standalone router.

RTP Real-Time Transport Protocol. Commonly used with IP networks. RTP is designed to provide end-to-end network transport functions for applications transmitting real-time data, such as audio, video, or simulation data, over multicast or unicast network services. RTP provides such services as payload type identification, sequence numbering, timestamping, and delivery monitoring to real-time applications.

RTT round-trip time. A measure of the time delay imposed by a network on a host from the sending of a packet until acknowledgement of the receipt.

RU rack unit. A rack is measured in rack units. An RU is equal to 44 mm or 1.75 inches.

S

SAP Signature Analysis Processor. Dispatches packets to the inspectors that are not stream-based and that are configured for interest in the packet in process.

SCEP Simple Certificate Enrollment Protocol. The Cisco Systems PKI communication protocol that leverages existing technology by using PKCS#7 and PKCS#10. SCEP is the evolution of the enrollment protocol.

SDEE Security Device Event Exchange. A product-independent standard for communicating security device events. It is an enhancement to RDEP. It adds extensibility features that are needed for communicating events generated by various types of security devices.

SDP Slave Dispatch Processor.

SEAF signature event action filter. Subtracts actions based on the signature event's signature ID, addresses, and RR. The input to the SEAF is the signature event with actions possibly added by the SEAO.

SEAH signature event action handler. Performs the requested actions. The output from SEAH is the actions being performed and possibly an <evIdsAlert> written to the Event Store.

SEAO signature event action override. Adds actions based on the RR value. SEAO applies to all signatures that fall into the range of the configured RR threshold. Each SEAO is independent and has a separate configuration value for each action type.

SEAP Signature Event Action Processor. Processes event actions. Event actions can be associated with an event risk rating (RR) threshold that must be surpassed for the actions to take place.

Secure Shell Protocol Protocol that provides a secure remote connection to a router through a Transmission Control Protocol (TCP) application.

Security Monitor Monitoring Center for Security. Provides event collection, viewing, and reporting capability for network devices. Used with the IDS MC.

sensing interface The interface on the sensor that monitors the desired network segment. The sensing interface is in promiscuous mode; it has no IP address and is not visible on the monitored segment.

sensor The sensor is the intrusion detection engine. It analyzes network traffic searching for signs of unauthorized activity.

SensorApp A component of the IPS. Performs packet capture and analysis. SensorApp analyzes network traffic for malicious content. Packets flow through a pipeline of processors fed by a producer designed to collect packets from the network interfaces on the sensor. Sensorapp is the standalone executable that runs Analysis Engine.

Service engine Deals with specific protocols, such as DNS, FTP, H255, HTTP, IDENT, MS RPC, MS SL. NTP, RPC, SMB, SNMP, and SSH.

service pack Used for the release of bug fixes with no new enhancements. Service packs are cumulative following a base version release (minor or major).

session command Command used on routers and switches to provide either Telnet or console access to a module in the router or switch.

shun command Enables a dynamic response to an attacking host by preventing new connections and disallowing packets from any existing connection. It is used by ARC when blocking with a PIX Firewall.

Signature Analysis Processor See SAP.

signature A signature distills network information and compares it against a rule set that indicates typical intrusion activity.

signature engine A component of the sensor that supports many signatures in a certain category. An engine is composed of a parser and an inspector. Each engine has a set of legal parameters that have allowable ranges or sets of values.

signature event action filter See SEAF.

signature event action handler See SEAH.

signature event action override See SEAO.

signature event action processor See SEAP.

signature update Executable image that updates the IPS signature analysis engine (SensorApp) and the NSDB. Applying an IPS signature update is like updating virus definitions on a virus scanning program. Signature updates are released independently and have their own versioning scheme.

Slave Dispatch Processor See SDP.

SMB Server Message Block. File-system protocol used in LAN manager and similar NOSs to package data and exchange information with other systems.

SMTP Simple Mail Transfer Protocol. Internet protocol providing e-mail services.

SN Serial Number. Part of the UDI. The SN is the serial number of your Cisco product.

sniffing interface See sensing interface.

SNMP Simple Network Management Protocol. Network management protocol used almost exclusively in TCP/IP networks. SNMP provides a means to monitor and control network devices, and to manage configurations, statistics collection, performance, and security.

SNMP2 SNMP Version 2. Version 2 of the network management protocol. SNMP2 supports centralized and distributed network management strategies, and includes improvements in the SMI, protocol operations, management architecture, and security.

software bypass Passes traffic through the IPS system without inspection.

source address Address of a network device that is sending data.

SP Statistics Processor. Keeps track of system statistics such as packet counts and packet arrival rates.

SPAN Switched Port Analyzer. Feature of the Catalyst 5000 switch that extends the monitoring abilities of existing network analyzers into a switched Ethernet environment. SPAN mirrors the traffic at one switched segment onto a predefined SPAN port. A network analyzer attached to the SPAN port can monitor traffic from any other Catalyst switched port.

spanning tree Loop-free subset of a network topology.

SQL Structured Query Language. International standard language for defining and accessing relational databases.

SRAM Type of RAM that retains its contents for as long as power is supplied. SRAM does not require constant refreshing, like DRAM

SRP Stream Reassembly Processor. Reorders TCP streams to ensure the arrival order of the packets at the various stream-based inspectors. It is also responsible for normalization of the TCP stream. The normalizer engine lets you enable or disable alert and deny actions.

SSH Secure Shell. A utility that uses strong authentication and secure communications to log in to another computer over a network.

SSL Secure Socket Layer. Encryption technology for the Internet used to provide secure transactions, such as the transmission of credit card numbers for e-commerce.

Stacheldraht A DDoS tool that relies on the ICMP protocol.

State engine Stateful searches of HTTP strings.

Statistics Processor See SP.

Stream Reassembly Processor See SRP.

String engine A signature engine that provides regular expression-based pattern inspection and alert functionality for multiple transport protocols, including TCP, UDP, and ICMP.

subsignature A more granular representation of a general signature. It typically further defines a broad scope signature.

surface mounting Refers to attaching rubber feet to the bottom of a sensor when it is installed on a flat surface. The rubber feet allow proper airflow around the sensor and they also absorb vibration so that the hard-disk drive is less impacted.

switch Network device that filters, forwards, and floods frames based on the destination address of each frame. The switch operates at the data link layer of the OSI model.

SYN flood Denial of Service attack that sends a host more TCP SYN packets (request to synchronize sequence numbers, used when opening a connection) than the protocol implementation can handle.

system image The full IPS application and recovery image used for reimaging an entire sensor.

T

TAC A Cisco Technical Assistance Center. There are four TACs worldwide.

TACACS+ Terminal Access Controller Access Control System Plus. Proprietary Cisco enhancement to Terminal Access Controller Access Control System (TACACS). Provides additional support for authentication, authorization, and accounting.

TCP Transmission Control Protocol. Connection-oriented transport layer protocol that provides reliable full-duplex data transmission. TCP is part of the TCP/IP protocol stack.

TCPDUMP The TCPDUMP utility is a free network protocol analyzer for UNIX and Windows. It lets you examine data from a live network or from a capture file on disk. You can use different options for viewing summary and detail information for each packet. For more information, go to http://www.tcpdump.org/.

TCP reset interface The interface on the IDS-4250-XL and IDSM-2 that can send TCP resets. On most sensors the TCP resets are sent out on the same sensing interface on which the packets are monitored, but on the IDS-4250-XL and IDSM-2 the sensing interfaces cannot be used for sending TCP resets. On the IDS-4250-XL the TCP reset interface is the onboard 10/100/100 TX interface, which is normally used on the IDS-4250-TX appliance when the XL card is not present. On the IDSM-2 the TCP reset interface is designated as port 1 with Catalyst software, and is not visible to the user in Cisco IOS software. The TCP reset action is only appropriate as an action selection on those signatures that are associated with a TCP-based service.

Telnet Standard terminal emulation protocol in the TCP/IP protocol stack. Telnet is used for remote terminal connection, enabling users to log in to remote systems and use resources as if they were connected to a local system. Telnet is defined in RFC 854.

terminal server A router with multiple, low speed, asynchronous ports that are connected to other serial devices. Terminal servers can be used to remotely manage network equipment, including sensors.

TFN2K Tribe Flood Network 2000. A common type of Denial of Service (DoS) attack that can take advantage of forged or rapidly changing source IP addresses to allow attackers to thwart efforts to locate or filter the attacks.

TFTP Trivial File Transfer Protocol. Simplified version of FTP that lets files be transferred from one computer to another over a network, usually without the use of client authentication (for example, username and password).

three-way handshake Process whereby two protocol entities synchronize during connection establishment.

threshold A value, either upper- or lower-bound that defines the maximum/minimum allowable condition before an alert is sent.

Time Processor See TP.

TLS Transport Layer Security. The protocol used over stream transports to negotiate the identity of peers and establish encrypted communications.

topology Physical arrangement of network nodes and media within an enterprise networking structure.

TP Time Processor. Processes events stored in a time-slice calendar. Its primary task is to make stale database entries expire and to calculate time-dependent statistics.

TPKT RFC 1006-defined method of demarking messages in a packet.

traceroute Program available on many systems that traces the path a packet takes to a destination. It is used mostly to debug routing problems between hosts. A traceroute protocol is also defined in RFC 1393.

traffic analysis Inference of information from observable characteristics of data flow(s), even when the data is encrypted or otherwise not directly available. Such characteristics include the identities and locations of the source(s) and destination(s), and the presence, amount, frequency, and duration of occurrence.

Traffic ICMP engine Analyzes traffic from nonstandard protocols, such as TFN2K, LOKI, and DDOS.

Transaction Server A component of the IPS.

Transaction Source A component of the IPS.

trap Message sent by an SNMP agent to an NMS, a console, or a terminal to indicate the occurrence of a significant event, such as a specifically defined condition or a threshold that was reached.

Trojan engine Analyzes traffic from nonstandard protocols, such as BO2K and TFN2K.

trunk Physical and logical connection between two switches across which network traffic travels. A backbone is composed of a number of trunks.

trusted certificate Certificate upon which a certificate user relies as being valid without the need for validation testing; especially a public-key certificate that is used to provide the first public key in a certification path.

trusted key Public key upon which a user relies; especially a public key that can be used as the first public key in a certification path.

tune Adjusting signature parameters to modify an existing signature.

U

UDI Unique Device Identifier. Provides a unique identity for every Cisco product. The UDI is composed of the PID, VID, and SN. The UDI is stored in the Cisco IPS ID PROM.

UDP User Datagram Protocol. Connectionless transport layer protocol in the TCP/IP protocol stack. UDP is a simple protocol that exchanges datagrams without acknowledgments or guaranteed delivery, requiring that error processing and retransmission be handled by other protocols. UDP is defined in RFC 768.

unblock To direct a router to remove a previously applied block.

UPS Uninterruptable Power Source.

UTC Coordinated Universal Time. Time zone at zero degrees longitude. Formerly called Greenwich Mean Time (GMT) and Zulu time.

V

VACL VLAN ACL. An ACL that filters all packets (both within a VLAN and between VLANs) that pass through a switch. Also known as security ACLs.

VID Version identifier. Part of the UDI.

VIP Versatile Interface Processor. Interface card used in Cisco 7000 and Cisco 7500 series routers. The VIP provides multilayer switching and runs Cisco IOS. The most recent version of the VIP is VIP2.

virtual sensor A logical grouping of sensing interfaces and the configuration policy for the signature engines and alarm filters to apply to them. In other words, multiple virtual sensors running on the same appliance, each configured with different signature behavior and traffic feeds. IPS 5.x supports only one virtual sensor.

virus Hidden, self-replicating section of computer software, usually malicious logic, that propagates by infecting—that is, inserting a copy of itself into and becoming part of—another program. A virus cannot run by itself; it requires that its host program be run to make the virus active.

virus update A signature update specifically addressing viruses.

VLAN Virtual Local Area Network. Group of devices on one or more LANs that are configured (using management software) so that they can communicate as if they were attached to the same wire, when in fact they are located on a number of different LAN segments. Because VLANs are based on logical instead of physical connections, they are extremely flexible.

VMS CiscoWorks VPN/Security Management Solution. A suite of network security applications that combines web-based tools for configuring, monitoring, and troubleshooting enterprise VPN, firewalls, network intrusion detection systems and host-based intrusion prevention systems.

VoIP Voice over IP. The capability to carry normal telephony-style voice over an IP-based internet with POTS-like functionality, reliability, and voice quality. VoIP enables a router to carry voice traffic (for example, telephone calls and faxes) over an IP network. In VoIP, the DSP segments the voice signal into frames, which then are coupled in groups of two and stored in voice packets. These voice packets are transported using IP in compliance with ITU-T specification H.323.

VPN Virtual Private Network(ing). Enables IP traffic to travel securely over a public TCP/IP network by encrypting all traffic from one network to another. A VPN uses "tunneling" to encrypt all information at the IP level.

vulnerability One or more attributes of a computer or a network that permit a subject to initiate patterns of misuse on that computer or network.

W

WAN wide-area network. Data communications network that serves users across a broad geographic area and often uses transmission devices provided by common carriers. Frame Relay, SMDS, and X.25 are examples of WANs.

Web Server A component of the IPS.

Wireshark Wireshark is a free network protocol analyzer for UNIX and Windows. It lets you examine data from a live network or from a capture file on disk. You can interactively browse the capture data, viewing summary and detail information for each packet. Wireshark has several powerful features, including a rich display filter language and the ability to view the reconstructed stream of a TCP session. For more information, go to http://www.wireshark.org.

worm A computer program that can run independently, can propagate a complete working version of itself onto other hosts on a network, and can consume computer resources destructively.

X

X.509 Standard that defines information contained in a certificate.

XML eXtensible Markup Language. Textual file format used for data interchange between heterogeneous hosts.

INDEX

Numerics

A

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

U

V

W