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
```

Cisco Intrusion Prevention System Sensor CLI Configuration Guide for IPS 5.1

```
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.