
# Configuring SNMP

This chapter describes how to configure SNMP, and contains the following sections:


- SNMP Notes and Caveats, page 15-1
- Understanding SNMP, page 15-1
- Configuring SNMP, page 15-2
- Configuring SNMPv3 Users, page 15-4
- Displaying SNMPv3 Users and Engine IDs, page 15-7
- Configuring SNMP Traps, page 15-8
- Supported MIBS, page 15-11


## SNMP Notes and Caveats

The following notes and caveats apply to SNMP:

- To have the sensor send SNMP traps, you must also select **request-snmp-trap** as the event action when you configure signatures.
- 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.
- 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.
- We recommend that you configure SNMPv3 users with security levels that require authentication, such as authPriv and authNoPriv, with authPriv being the most highly recommended. Configuring SNMPv3 users with the noAuthNoPriv security level is NOT recommended.

## 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.

You can use SNMPv2 and SNMPv3 protocol concurrently. If the SNMP request contains version 3 user information, then you get a version 3 reply (provided the same user is configured as a version 3 user in the IPS). If the SNMP request is a version 2 request, the IPS returns the response (provided the correct version 2 community string is configured).

**Note** Encryption of the SNMPv3 payload uses AES-128 and authentication of the user password uses HMAC-SHA-96.

# Configuring SNMP

**Caution** To have the sensor send SNMP traps, you must also select **request-snmp-trap** as the event action when you configure signatures.

Configure general SNMP parameters in the service notification submode.

The following commands 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**—Removes an entry or selection setting.
- **read-only-community**—Specifies the read-only community name for the SNMP agent. The default is public.
- **read-write-community**—Specifies the read-write community name for the SNMP agent. The default is private.

- **snmp-agent-port**—Specifies the port the SNMP agent will listen on. The default SNMP port number is 161.
- **snmp-agent-protocol**—Specifies the protocol the SNMP agent will communicate with. The default protocol is UDP.
- **system-contact**—Specifies the contact information for this sensor. The system-contact option modifies the SNMPv2-MIB::sysContact.0 value.
- **system-location**—Specifies the location of the sensor. The system-location option modifies the SNMPv2-MIB::sysLocation.0 value.

### Configuring SNMP General Parameters

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** Specify 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. The read-only community name specifies the password for queries to the SNMP agent.

```
sensor(config-not)# read-only-community PUBLIC1
```

**b.** Assign the read-write community string. The read-write community name specifies the password for **sets** to the SNMP agent.

```
sensor(config-not)# read-write-community PRIVATE1
```

**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.** Enter the location of the sensor.

```
sensor(config-not)# system-location AUSTIN
```

**e.** Enter 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. Specify 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.

### For More Information

For the procedure for assigning actions to signatures, see Assigning Actions to Signatures, page 7-20.

# Configuring SNMPv3 Users

You configure SNMP v3users in the service notification submode.

The following parameters apply:

- **v3users [user-name]**—Lets you add SNMPv3 users. The maximum number of characters for the user name is 32.
- **access-control**—Sets the access privilege for the SNMPv3 user:
  - **rouser**—User with read-only access.
  - **rwuser**—User with read-write access.

**Note** Both rouser and rwuser can do ‘get’ operations, but rwuser access control is mandatory to do ‘set’ operations.

- **security-level** [**noAuthNoPriv | authNoPriv | authPriv**]—Sets the security level for the SNMPv3 user:
  - **noAuthNoPriv**—No Authentication and No Privacy, which means that no security is applied to messages.
  - **authNoPriv**—Authentication but No Privacy, which means that messages are authenticated.
  - **authPriv**—Authentication and Privacy, which means that messages are authenticated and encrypted.

**Note** We recommend that you configure SNMPv3 users with security levels that require authentication, such as authPriv and authNoPriv, with authPriv being the most highly recommended. Configuring SNMPv3 users with the noAuthNoPriv security level is NOT recommended.

**Note** For SNMPv3 users with authPriv and authNoPriv security levels, you must specify all security parameters or the changes are not applied and you receive a warning message.

- **auth-protocol [SHA]**—Sets the authentication protocol for the SNMPv3 user to SHA. If you do not set the authentication protocol to SHA, the default is none.
- **auth-passphrase** *passphrase*—Lets you enter an authentication passphrase for the SNMPv3 user. The valid range is 8 to 257 characters; no spaces or double quotes allowed.
- **priv-protocol [AES]**—Sets the encryption protocol to AES. If you do not set the privacy protocol, the default is none.
- **priv-passphrase** *passphrase*—Lets you enter an encryption passphrase for the SNMPv3 user. The valid range is 8 to 257 characters; no spaces or double quotes allowed.

**Caution** The same passphrase value for both authentication and privacy is allowed, although it is not a recommended security practice.

- **default**—Sets the value back to the system default setting.
- **no**—Removes an entry or selection setting.

### Configuring SNMPv3 Users

To configure SNMPv3 users, 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** Add an SNMPv3 user.

```
sensor(config-not)# v3users user1
sensor(config-not-v3u)
```

**Step 4** Set the access privilege for this user:

```
sensor(config-not-v3u)# access-control rwuser
sensor(config-not-v3u)
```

**Step 5** Set the security level for this user.

```
sensor(config-not-v3u)# security-level authPriv
sensor(config-not-v3u)
```

**Step 6** Set the authentication protocol to SHA.

```
sensor(config-not-v3u)# auth-protocol SHA
sensor(config-not-v3u)
```

**Step 7** Enter an authentication passphrase for the SNMPv3 user and confirm it.

```
sensor(config-not-v3u)# auth-passphrase
Enter auth-passphrase[]: ********
Re-enter auth-passphrase: ********
sensor(config-not-v3u)#
```

**Step 8** Set the privacy protocol to AES.

```
sensor(config-not-v3u)# priv-protocol AES
sensor(config-not-v3u)#
```

**Step 9** Enter a privacy passphrase for the SNMPv3 user and confirm it.

```
sensor(config-not-v3u)# priv-passphrase
Enter priv-passphrase[]: *********
Re-enter priv-passphrase: *********
sensor(config-not-v3u)#
```

**Step 10** Verify the settings.

```
sensor(config-not-v3u)# show settings
   user-name: user1
   -----------------------------------------------
      access-control: rwuser default: rouser
      security-level: authPriv default: noAuthNoPriv
      auth-protocol: SHA default:
      priv-protocol: AES default:
      auth-passphrase: <hidden>
      priv-passphrase: <hidden>
   -----------------------------------------------
sensor(config-not-v3u)#
```

**Step 11** Exit notification submode.

```
sensor(config-not-v3u)# exit
sensor(config-not)# exit
Apply Changes:?[yes]:
```

**Step 12** Press **Enter** to apply the changes or enter **no** to discard them.

# Displaying SNMPv3 Users and Engine IDs

You need to know which SNMPv3 users are configured on the sensor so you can add them to SNMP traps. Every SNMPv3 agent has an engine ID that serves as a unique identifier for the agent. The engine ID is used with a hashing function to generate keys for authentication and encryption of SNMP v3 messages. Use the **show snmpv3** users and **show snmpv3 engineID** commands in EXEC mode to display users and engine IDs.

To display SNMPv3 users and engine IDs, follow these steps:

**Step 1** Log in to the CLI.

**Step 2** Display the SNMPv3 users configured on the sensor.

```
sensor# show snmpv3 users
Username: cisco1233333
Engine ID: 80001f88034c4e35ea727f
Access Control: rouser
Security Level: authNoPriv
Auth Protocol: SHA
------------------------------------------------
Username: cisco1234
Engine ID: 80001f88034c4e35ea727f
Access Control: rouser
Security Level: authNoPriv
Auth Protocol: SHA
------------------------------------------------
Username: cisco12344
Engine ID: 80001f88034c4e35ea727f
Access Control: rouser
Security Level: noAuthNoPriv
------------------------------------------------
Username: cisco1234444
Engine ID: 80001f88034c4e35ea727f
Access Control: rouser
Security Level: authPriv
Auth Protocol: SHA
Priv Protocol: AES
------------------------------------------------
Username: cisco123456
Engine ID: 80001f88034c4e35ea727f
Access Control: rwuser
Security Level: authPriv
Auth Protocol: SHA
Priv Protocol: AES
sensor#
```

**Step 3** Display the SNMPv3 Engine ID.

```
sensor# show snmpv3 engineID
Local SNMP engineID: 80001f88034c4e35ea727f
sensor#
```

**Step 4** Display the SNMPv3 users associated with a trap destination.

```
sensor# show snmpv3 users
Username: sat
Engine ID: 80001f8803503de59e5831
Access Control: rouser
Security Level: noAuthNoPriv
Associated with Trap Destination(s): 10.10.10.10:162
sensor#
```

# Configuring SNMP Traps

You can configure SNMP traps in the service notification submode. An SNMP trap is a notification. You configure the sensor to send traps based on whether the event is fatal, an error, or a warning.

You can also associate SNMPv3 users with SNMP trap destinations. If no SNMPv3 user is associated with a trap, then an SNMPv2 trap is sent. For example, if a version 3 user is associated with a trap destination, all traps for that destination will be version 3 traps using the configured user. No version 2 trap is sent to that trap destination. If a version 3 user is not configured, then a version 2 trap is sent. Traps can be sent to one destination using version 3 and to another destination using version 2. Support for SNMPv3 is valid for IPS 7.2(2)E4 and later.

The following parameters 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-health-traps {true | false}** —Enables the sending of both heartbeat and health metric change traps.

  **Note** To receive sensor health information through SNMP traps, you must have the sensor health metrics enabled. Use the **heartbeat-events enable** command in service health monitor submode to enable sensor health metrics.

- **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**—Specifies the community name used when sending traps. If no community name is specified the general trap community name is used.
- **trap-destinations**—Defines the destinations to send error events and alert events generated from signature actions:
  - **trap-community-name**—Specifies the community name used when sending the trap. If no community name is specified the general trap community name is used.
  - **trap-port**—Specifies the port number to send the SNMP trap to.
  - **trap-v3user**—Specifies the SNMPv3 user of the trap destination. If no SNMPv3 user is specified, SNMPv2 is used.

**Note** You must save all configuration changes for an SNMPv3 user to be associated with the trap destination.

**Note** You cannot remove an SNMPv3 user associated with a trap destination unless you disassociate the user from the trap destination.

### Configuring SNMP Traps

To configure SNMP traps and add SNMPv3 users to the 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** Specify the parameters for the SNMP trap:

**a.** Specify 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.** Specify whether you want detailed SNMP traps.

```
sensor(config-not)# enable-detail-traps true
```

**c.** Specify whether you want health traps (heartbeat and health metric change traps).

```
sensor(config-not)# enable-health-traps true
```

**Note** Make sure heartbeat and sensor health metrics are enabled.

**d.** Enter the community string to be included in the detailed traps.

```
sensor(config-not)# trap-community-name TRAP1
```

**Step 5** Specify the parameters for the SNMP trap destinations so the sensor knows which management workstations to send them to:

**a.** Enter the IP address of the SNMP management station.

```
sensor(config-not)# trap-destinations 10.0.0.0
```

**b.** Enter the UDP port of the SNMP management station. The default is 162

```
sensor(config-not-tra)# trap-port 162
```

**c.** Enter the trap community string.

```
sensor(config-not-tra)# trap-community-name AUSTIN_PUBLIC1
```

**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.

**d.** Add an SNMPv3 user. Check first to see what SNMPv3 users have been configured on the sensor. If no SNMPv3 users have been configured on the sensor, SNMPv2 is used.

```
sensor(config-not-tra)# trap-v3user ?
cisco1233333     Configured SNMPv3 User
cisco1234        Configured SNMPv3 User
cisco12344       Configured SNMPv3 User
cisco1234444     Configured SNMPv3 User
cisco123456      Configured SNMPv3 User
sensor(config-not-tra)# trap-v3user cisco1234
sensor(config-not-tra)#
```

**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: 10.0.0.0
      -----------------------------------------------
         trap-community-name: AUSTIN_PUBLIC1 default:
         trap-port: 162 default: 162
         trap-v3user: cisco1234 default:
      -----------------------------------------------
   -----------------------------------------------
   error-filter: warning|error|fatal default: error|fatal
   enable-detail-traps: true default: false
   enable-health-traps: true default: false
   enable-notifications: true default: false
   enable-set-get: false <defaulted>
   snmp-agent-port: 161 <defaulted>
   snmp-agent-protocol: udp <defaulted>
   read-only-community:  default:
   read-write-community:  default:
   trap-community-name: TRAP1 default: public
   system-location: Unknown <defaulted>
   system-contact: Unknown <defaulted>
   v3users (min: 0, max: 25, current: 5)
   -------------------------------------------------
      user-name: cisco1233333
      -------------------------------------------------
         access-control: rouser <defaulted>
         security-level: authNoPriv default: noAuthNoPriv
         auth-protocol: SHA default:
         priv-protocol:  default:
         auth-passphrase: <hidden>
         priv-passphrase: <hidden>
      -------------------------------------------------
      user-name: cisco1234
      -------------------------------------------------
         access-control: rouser <defaulted>
         security-level: authNoPriv default: noAuthNoPriv
```

```
            auth-protocol: SHA default:
            priv-protocol:  default:
            auth-passphrase: <hidden>
            priv-passphrase: <hidden>
         -----------------------------------------------
         user-name: cisco12344
         -----------------------------------------------
            access-control: rouser <defaulted>
            security-level: noAuthNoPriv <defaulted>
            auth-protocol:  default:
            priv-protocol:  default:
            auth-passphrase: <hidden>
            priv-passphrase: <hidden>
         -----------------------------------------------
         user-name: cisco1234444
         -----------------------------------------------
            access-control: rouser <defaulted>
            security-level: authPriv default: noAuthNoPriv
            auth-protocol: SHA default:
            priv-protocol: AES default:
            auth-passphrase: <hidden>
            priv-passphrase: <hidden>
         -----------------------------------------------
         user-name: cisco123456
         -----------------------------------------------
            access-control: rwuser default: rouser
            security-level: authPriv default: noAuthNoPriv
            auth-protocol: SHA default:
            priv-protocol: AES default:
            auth-passphrase: <hidden>
            priv-passphrase: <hidden>
         -----------------------------------------------
      -----------------------------------------------
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.

---

#### For More Information

For the procedure for assigning actions to signatures, see Assigning Actions to Signatures, page 7-20.

# Supported MIBS

The following private MIBs are supported on the sensor:

- CISCO-CIDS-MIB

  The CISCO-CIDS-MIB has been updated to include SNMP health data.
- 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.