SNMP Configuration

• SNMP Overview, on page 1
• SNMP Functional Overview, on page 1
• SNMP Notifications, on page 2
• SNMP Security Levels and Privileges, on page 2
• Supported Combinations of SNMP Security Models and Levels, on page 3
• SNMPv3 Security Features, on page 3
• SNMP Support, on page 3
• Configuring SNMP, on page 4

SNMP Overview

The Simple Network Management Protocol (SNMP) is an application-layer protocol that provides a message format for communication between SNMP managers and agents. SNMP provides a standardized framework and a common language for monitoring and managing devices in a network.

SNMP Functional Overview

The SNMP framework consists of three parts:

• An SNMP manager—The system used to control and monitor the activities of network devices using SNMP.

• An SNMP agent—The software component within Cisco UCS, the managed device that maintains the data for Cisco UCS, and reports the data as needed to the SNMP manager. Cisco UCS includes the agent and a collection of MIBs. To enable the SNMP agent and create the relationship between the manager and agent, enable and configure SNMP in Cisco UCS Manager.

• A managed information base (MIB)—The collection of managed objects on the SNMP agent. Cisco UCS release 1.4(1) and higher supports a larger number of MIBs than earlier releases.

Cisco UCS supports SNMPv1, SNMPv2c and SNMPv3. Both SNMPv1 and SNMPv2c use a community-based form of security. SNMP is defined in the following:

• RFC 3410 (http://tools.ietf.org/html/rfc3410)

• RFC 3411 (http://tools.ietf.org/html/rfc3411)
SNMP Notifications

A key feature of SNMP is the ability to generate notifications from an SNMP agent. These notifications do not require that requests be sent from the SNMP manager. Notifications can indicate improper user authentication, restarts, the closing of a connection, loss of connection to a neighbor router, or other significant events.

Cisco UCS Manager generates SNMP notifications as either traps or informs. Traps are less reliable than informs because the SNMP manager does not send any acknowledgment when it receives a trap, and Cisco UCS Manager cannot determine if the trap was received. An SNMP manager that receives an inform request acknowledges the message with an SNMP response Protocol Data Unit (PDU). If the Cisco UCS Manager does not receive the PDU, it can send the inform request again.

SNMP Security Levels and Privileges

SNMPv1, SNMPv2c, and SNMPv3 each represent a different security model. The security model combines with the selected security level to determine the security mechanism applied when the SNMP message is processed.

The security level determines the privileges required to view the message associated with an SNMP trap. The privilege level determines whether the message requires protection from disclosure or whether the message is authenticated. The supported security level depends on which security model is implemented. SNMP security levels support one or more of the following privileges:

- noAuthNoPriv—No authentication or encryption
- authNoPriv—Authentication but no encryption
- authPriv—Authentication and encryption

SNMPv3 provides for both security models and security levels. A security model is an authentication strategy that is set up for a user and the role in which the user resides. A security level is the permitted level of security within a security model. A combination of a security model and a security level determines which security mechanism is employed when handling an SNMP packet.
Supported Combinations of SNMP Security Models and Levels

The following table identifies the combinations of security models and levels.

<table>
<thead>
<tr>
<th>Model</th>
<th>Level</th>
<th>Authentication</th>
<th>Encryption</th>
<th>What Happens</th>
</tr>
</thead>
<tbody>
<tr>
<td>v1</td>
<td>noAuthNoPriv</td>
<td>Community string</td>
<td>No</td>
<td>Uses a community string match for authentication.</td>
</tr>
<tr>
<td>v2c</td>
<td>noAuthNoPriv</td>
<td>Community string</td>
<td>No</td>
<td>Uses a community string match for authentication.</td>
</tr>
<tr>
<td>v3</td>
<td>noAuthNoPriv</td>
<td>Username</td>
<td>No</td>
<td>Uses a username match for authentication.</td>
</tr>
<tr>
<td>v3</td>
<td>authNoPriv</td>
<td>HMAC-MD5 or HMAC-SHA</td>
<td>No</td>
<td>Provides authentication based on the Hash-Based Message Authentication Code (HMAC) Message Digest 5 (MD5) algorithm or the HMAC Secure Hash Algorithm (SHA).</td>
</tr>
<tr>
<td>v3</td>
<td>authPriv</td>
<td>HMAC-MD5 or HMAC-SHA</td>
<td>DES</td>
<td>Provides authentication based on the HMAC-MD5 or HMAC-SHA algorithms. Provides Data Encryption Standard (DES) 56-bit encryption in addition to authentication based on the Cipher Block Chaining (CBC) DES (DES-56) standard.</td>
</tr>
</tbody>
</table>

SNMPv3 Security Features

SNMPv3 provides secure access to devices through a combination of authenticating and encrypting frames over the network. SNMPv3 authorizes only configured users to perform management operations and encrypts SNMP messages. The SNMPv3 User-Based Security Model (USM) refers to SNMP message-level security and offers the following services:

- Message integrity—Ensures that messages are not altered or destroyed in an unauthorized manner, and that data sequences are not altered beyond what can occur non-maliciously.

- Message origin authentication—Ensures that the identity of a message originator is verifiable.

- Message confidentiality and encryption—Ensures that information is not made available or disclosed to unauthorized individuals, entities, or processes.

SNMP Support

Cisco UCS provides the following support for SNMP:

**Support for MIBs**

Cisco UCS supports read-only access to MIBs.

**Configuration Protocols for SNMPv3 Users**

Cisco UCS supports the following authentication protocols for SNMPv3 users:

- HMAC-MD5-96 (MD5)
- HMAC-SHA-96 (SHA)

**AES Privacy Protocol for SNMPv3 Users**

Cisco UCS uses Advanced Encryption Standard (AES) as one of the privacy protocols for SNMPv3 message encryption and conforms with RFC 3826.

The privacy password, or priv option, offers a choice of DES or 128-bit AES encryption for SNMP security encryption. If you enable AES-128 configuration and include a privacy password for an SNMPv3 user, Cisco UCS Manager uses the privacy password to generate a 128-bit AES key. The AES privacy password can have a minimum of eight characters. If the passphrases are specified in clear text, you can specify a maximum of 64 characters.

### Configuring SNMP

#### Enabling SNMP and Configuring SNMP Properties

SNMP messages from a Cisco UCS domain display the fabric interconnect name rather than the system name.

**Procedure**

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>UCS-A# scope monitoring</td>
<td>Enters monitoring mode.</td>
</tr>
<tr>
<td>Step 2</td>
<td>UCS-A/monitoring # enable snmp</td>
<td>Enables SNMP.</td>
</tr>
<tr>
<td>Step 3</td>
<td>UCS-A/monitoring # set snmp community</td>
<td>Enters snmp community mode.</td>
</tr>
<tr>
<td>Step 4</td>
<td>UCS-A/monitoring # Enter a snmp</td>
<td>Specifies SNMP community. Use the community name as a password. The</td>
</tr>
<tr>
<td></td>
<td>community: community-name</td>
<td>community name can be any alphanumeric string up to 32 characters.</td>
</tr>
<tr>
<td>Step 5</td>
<td>UCS-A/monitoring # set snmp syscontact</td>
<td>Specifies the system contact person responsible for the SNMP. The system</td>
</tr>
<tr>
<td></td>
<td>system-contact-name</td>
<td>contact name can be any alphanumeric string up to 255 characters,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>such as an email address or name and telephone number.</td>
</tr>
</tbody>
</table>
### Example

The following example enables SNMP, configures an SNMP community named SnmpCommSystem2, configures a system contact named contactperson, configures a contact location named systemlocation, and commits the transaction:

```
UCS-A# scope monitoring
UCS-A /monitoring # enable snmp
UCS-A /monitoring* # set snmp community
UCS-A /monitoring* # Enter a snmp community: SnmpCommSystem2
UCS-A /monitoring* # set snmp syscontact contactperson1
UCS-A /monitoring* # set snmp syslocation systemlocation
UCS-A /monitoring* # commit-buffer
UCS-A /monitoring #
```

### What to do next

Create SNMP traps and users.

## Creating an SNMP Trap

### Procedure

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> UCS-A# scope monitoring</td>
<td>Enters monitoring mode.</td>
</tr>
<tr>
<td><strong>Step 2</strong> UCS-A /monitoring # enable snmp</td>
<td>Enables SNMP.</td>
</tr>
<tr>
<td><strong>Step 3</strong> UCS-A /monitoring # create snmp-trap {hostname</td>
<td>ip-addr</td>
</tr>
<tr>
<td><strong>Step 4</strong> UCS-A /monitoring/snmp-trap # set community community-name</td>
<td>Specifies the SNMP community name to be used for the SNMP trap.</td>
</tr>
<tr>
<td><strong>Step 5</strong> UCS-A /monitoring/snmp-trap # set port port-num</td>
<td>Specifies the port to be used for the SNMP trap.</td>
</tr>
<tr>
<td><strong>Step 6</strong> UCS-A /monitoring/snmp-trap # set version {v1</td>
<td>v2c</td>
</tr>
</tbody>
</table>
### Creating an SNMP Trap

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 7</strong> (Optional) UCS-A /monitoring/snmp-trap # set notificationtype {traps</td>
<td>informs}</td>
</tr>
<tr>
<td></td>
<td>• traps—SNMP trap notifications</td>
</tr>
<tr>
<td></td>
<td>• informs—SNMP inform notifications</td>
</tr>
<tr>
<td><strong>Step 8</strong> (Optional) UCS-A /monitoring/snmp-trap # set v3 privilege {auth</td>
<td>noauth</td>
</tr>
<tr>
<td></td>
<td>• auth—Authentication but no encryption</td>
</tr>
<tr>
<td></td>
<td>• noauth—No authentication or encryption</td>
</tr>
<tr>
<td></td>
<td>• priv—Authentication and encryption</td>
</tr>
<tr>
<td><strong>Step 9</strong> UCS-A /monitoring/snmp-trap # commit-buffer</td>
<td>Commits the transaction to the system configuration.</td>
</tr>
</tbody>
</table>

### Example

The following example enables SNMP, creates an SNMP trap using an IPv4 address, specifies that the trap will use the SnmpCommSystem2 community on port 2, sets the version to v3, sets the notification type to traps, sets the v3 privilege to priv, and commits the transaction:

```
UCS-A# scope monitoring
UCS-A /monitoring # enable snmp
UCS-A /monitoring* # create snmp-trap 100.10.111.112
UCS-A /monitoring/snmp-trap* # set community SnmpCommSystem2
UCS-A /monitoring/snmp-trap* # set port 2
UCS-A /monitoring/snmp-trap* # set version v3
UCS-A /monitoring/snmp-trap* # set notificationtype traps
UCS-A /monitoring/snmp-trap* # set v3 privilege priv
UCS-A /monitoring/snmp-trap* # commit-buffer
```

The following example enables SNMP, creates an SNMP trap using an IPv6 address, specifies that the trap will use the SnmpCommSystem3 community on port 2, sets the version to v3, sets the notification type to traps, sets the v3 privilege to priv, and commits the transaction:

```
UCS-A# scope monitoring
UCS-A /monitoring # enable snmp
UCS-A /monitoring* # create snmp-trap 2001::1
UCS-A /monitoring/snmp-trap* # set community SnmpCommSystem3
UCS-A /monitoring/snmp-trap* # set port 2
UCS-A /monitoring/snmp-trap* # set version v3
UCS-A /monitoring/snmp-trap* # set notificationtype traps
UCS-A /monitoring/snmp-trap* # set v3 privilege priv
UCS-A /monitoring/snmp-trap* # commit-buffer
```
Deleting an SNMP Trap

Procedure

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>UCS-A# scope monitoring</td>
<td>Enters monitoring mode.</td>
</tr>
<tr>
<td>Step 2</td>
<td>UCS-A /monitoring # delete snmp-trap {hostname</td>
<td>ip-addr}</td>
</tr>
<tr>
<td>Step 3</td>
<td>UCS-A /monitoring # commit-buffer</td>
<td>Commits the transaction to the system configuration.</td>
</tr>
</tbody>
</table>

Example

The following example deletes the SNMP trap at IP address 192.168.100.112 and commits the transaction:

UCS-A# scope monitoring
UCS-A /monitoring # delete snmp-trap 192.168.100.112
UCS-A /monitoring* # commit-buffer
UCS-A /monitoring #

Creating an SNMPv3 User

Procedure

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>UCS-A# scope monitoring</td>
<td>Enters monitoring mode.</td>
</tr>
<tr>
<td>Step 2</td>
<td>UCS-A /monitoring # enable snmp</td>
<td>Enables SNMP.</td>
</tr>
<tr>
<td>Step 3</td>
<td>UCS-A /monitoring # create snmp-user user-name</td>
<td>Creates the specified SNMPv3 user.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>An SNMP username cannot be the same as a local username. Choose an SNMP username that does not match a local username.</td>
</tr>
<tr>
<td>Step 4</td>
<td>UCS-A /monitoring/snmp-user # set aes-128 {no</td>
<td>yes}</td>
</tr>
<tr>
<td>Step 5</td>
<td>UCS-A /monitoring/snmp-user # set auth {md5</td>
<td>sha}</td>
</tr>
<tr>
<td>Step 6</td>
<td>UCS-A /monitoring/snmp-user # set password</td>
<td>Specifies the user password. After you enter the set password command, you are prompted to enter and confirm the password.</td>
</tr>
<tr>
<td>Step 7</td>
<td>UCS-A /monitoring/snmp-user # set priv-password</td>
<td>Specifies the user privacy password. After you enter the set priv-password command, you...</td>
</tr>
</tbody>
</table>
Deleting an SNMPv3 User

### Example

The following example enables SNMP, creates an SNMPv3 user named snmp-user14, disables AES-128 encryption, specifies the use of MD5 authentication, sets the password and privacy password, and commits the transaction:

```
UCS-A# scope monitoring
UCS-A /monitoring # enable snmp
UCS-A /monitoring* # create snmp-user snmp-user14
UCS-A /monitoring/snmp-user* # set aes-128 no
UCS-A /monitoring/snmp-user* # set auth md5
UCS-A /monitoring/snmp-user* # set password
Enter a password:
Confirm the password:
UCS-A /monitoring/snmp-user* # set priv-password
Enter a password:
Confirm the password:
UCS-A /monitoring/snmp-user* # commit-buffer
UCS-A /monitoring/snmp-user #
```

### Deleting an SNMPv3 User

#### Procedure

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> UCS-A# scope monitoring</td>
<td>Enters monitoring mode.</td>
</tr>
<tr>
<td><strong>Step 2</strong> UCS-A /monitoring # delete snmp-user user-name</td>
<td>Deletes the specified SNMPv3 user.</td>
</tr>
<tr>
<td><strong>Step 3</strong> UCS-A /monitoring # commit-buffer</td>
<td>Commits the transaction to the system configuration.</td>
</tr>
</tbody>
</table>

#### Example

The following example deletes the SNMPv3 user named snmp-user14 and commits the transaction:

```
UCS-A# scope monitoring
UCS-A /monitoring # delete snmp-user snmp-user14
UCS-A /monitoring* # commit-buffer
UCS-A /monitoring #
```