Configuring Communication Services

This chapter includes the following sections:

Enabling or Disabling TLS v1.2

Beginning with release 4.2(2a), Cisco IMC supports disabling TLS v1.2 and also customize the cipher values for both v1.2 and v1.3.

Before you begin

If **CC** (Common Criteria) under **Security Configuration** is enabled, you cannot disable TLS v1.2. Ensure that **CC** is disabled before you disable TLS v1.2.

Enabling or disabling TLS v1.2, restarts vKVM, Webserver, XML API, and Redfish API sessions.

SUMMARY STEPS

1. Server# **scope cimc**
2. Server# **scope tls-config**
3. Server/tls-config # **set tlsv2Enabled** *yes*|*no*
4. Server/tls-config* # **Commit**
5. Server/tls-config # **set tlsv2CipherMode** *Custom*|*High*|*Low*|*Medium*
6. (Optional) Server/tls-config # **set tlsv2CipherMode Custom** *Cipher_Value*
7. Server/tls-config* # **Commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	Server# **scope cimc**	
Step 2	Server# **scope tls-config**	Enters the TLS configuration mode.
Step 3	Server/tls-config # **set tlsv2Enabled** *yes*|*no*	Enter **y** to confirm. Enables or Disables TLS v1.2.
Step 4	Server/tls-config* # **Commit**	Saves the changes.
Step 5	Server/tls-config # **set tlsv2CipherMode** *Custom*|*High*|*Low*|*Medium*	Selecting **High**, **Low**, or **Medium** automatically provides preset cipher values.
Step 6	(Optional) Server/tls-config # **set tlsv2CipherMode Custom** *Cipher_Value*	Enter a valid cipher value for **Custom** cipher mode. **Note** Refer https://www.openssl.org/docs/man1.0.2/man1/ciphers.html for OpenSSL equivalent cipher name for a specific cipher to be provided in custom cipher. If the cipher value entered is invalid or unsupported, then while saving the configuration, Cisco IMC automatically changes the **TLS v1.2 Cipher Mode** value to **High** and saves the configuration. You may see the following status: `TLS v1.2 Custom Cipher Status: Error: Configuring an invalid or unsupported TLS v1.2 Cipher List-'Cipher_Name'. Setting TLS v1.2 Cipher Mode to High.`
Step 7	Server/tls-config* # **Commit**	Saves the changes.

Example

Following example shows how to enable TLS v1.2 and set cipher mode to high:

```
Server# scope cimc
Server /cimc # scope tls-config
Server /cimc/tls-config # set tlsv2Enabled yes
Server /cimc/tls-config* # commit
Server /cimc/tls-config # set tlsv2CipherMode high
Server /cimc/tls-config* # commit
```

Following example shows how to enable TLS v1.2 and set cipher mode to custom:

```
server# scope cimc
server /cimc # scope tls-config
server /cimc/tls-config # set tlsv2CipherMode Custom
server /cimc/tls-config *# set tlsv2CipherList ECDHE-RSA-AES256-GCM-SHA384
server /cimc/tls-config *# commit
```

Enabling TLS Static Key Cipher

Perform this procedure to enable TLS static key cipher for Cisco UCS servers. TLS static key cipher is disabled by default.

Note You can enable this feature only through Cisco IMC CLI interface.

Static key cipher option is not applicable when **TLS v1.2 Cipher Mode** is set to **High** or **Custom**.

Static key cipher, if enabled, switches to NA automatically when **TLS v1.2 Cipher Mode** changes from **Medium**/**Low** to **High**/**Custom**.

SUMMARY STEPS

1. Server# **scope cimc**
2. Server /chassis # **scope tls-config**
3. Server /chassis/tls-config # **show detail**
4. Server /chassis/tls-config # **set static-cipher-enabled yes**
5. Server /chassis/tls-config # **commit**
6. Type **y** and press **Enter**.

DETAILED STEPS

	Command or Action	Purpose
Step 1	Server# **scope cimc**	Enters the Cisco IMC command mode.
Step 2	Server /chassis # **scope tls-config**	Enters the TLS configuration mode.
Step 3	Server /chassis/tls-config # **show detail**	Displays the **TLS Static Cipher Enabled** status: `TLS Configuration : TLS Static Cipher Enabled: no`
Step 4	Server /chassis/tls-config # **set static-cipher-enabled yes**	Enables TLS cipher.
Step 5	Server /chassis/tls-config # **commit**	Following warning is displayed. `Warning: This will enable static ciphers in TLS. KVM, Webserver, XMLAPI and Redfish sessions will be disconnected. Do you wish to continue? [[Y]es/[N]o]`
Step 6	Type **y** and press **Enter**.	Commits the transaction to the system configuration.

Example

This example shows how to enable TLS static key cipher:

```
Server# scope cimc
Server /cimc # scope tls-config
Server /cimc/tls-config # show detail
TLS Configuration :
    TLS Static Cipher Enabled: no
Server /cimc/tls-config #
Server /cimc/tls-config # set static-cipher-enabled yes
Server /cimc/tls-config *# commit
Warning: This will enable static ciphers in TLS.
         KVM, Webserver, XMLAPI and Redfish sessions will be disconnected.
Do you wish to continue? [[Y]es/[N]o] y
Server /cimc/tls-config # show detail
TLS Configuration :
    TLS Static Cipher Enabled: yes
```

Configuring HTTP

Beginning with release 4.1(2b), Cisco IMC supports separate HTTPS and HTTP communication services. You can disable only HTTP services using this functionality.

This functionality is supported only on the following servers:

- Cisco UCS C220 M5
- Cisco UCS C240 M5
- Cisco UCS C480 M5
- Cisco UCS C480 ML M5
- Cisco UCS C240 SD M5
- Cisco UCS C125 M5
- Cisco UCS S3260 M4/M5

Note If **Redirect HTTP to HTTPS Enabled** was disabled in any release earlier than 4.1(2b), then after upgrading to release 4.1(2b) or later, **HTTP Enabled** value is set to **Disabled** by the system.

Before you begin

You must log in as a user with admin privileges to configure HTTP.

SUMMARY STEPS

1. Server# **scope http**
2. Server /http # **set https-enabled** {**yes** | **no**}
3. Server /http # **set http-enabled** {**yes** | **no**}
4. Server /http # **set http-port** *number*
5. Server /http # **set https-port** *number*
6. Server /http # **set http-redirect** {**yes** | **no**}
7. Server /http # **set timeout** *seconds*

8. Server /http # **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	Server# **scope http**	Enters the HTTP command mode.
Step 2	Server /http # **set https-enabled** {**yes** | **no**}	Enables the HTTPS services or disables both HTTPS and HTTP services on Cisco IMC.
Step 3	Server /http # **set http-enabled** {**yes** | **no**}	Enables or disables HTTP services on the Cisco IMC.
Step 4	Server /http # **set http-port** *number*	Sets the port to use for HTTP communication. The default is 80.
Step 5	Server /http # **set https-port** *number*	Sets the port to use for HTTPS communication. The default is 443.
Step 6	Server /http # **set http-redirect** {**yes** | **no**}	**Note** This option is applicable only when HTTP is enabled. Enables or disables the redirection of an HTTP request to HTTPS.
Step 7	Server /http # **set timeout** *seconds*	Sets the number of seconds to wait between HTTP requests before the times out and terminates the session. Enter an integer between 60 and 10,800. The default is 1,800 seconds.
Step 8	Server /http # **commit**	Commits the transaction to the system configuration.

Example

This example configures HTTP for the Cisco IMC:

```
Server# scope http
Server /http # set https-enabled yes
Server /http # set http-enabled yes
Server /http *# set http-port 80
Server /http *# set https-port 443
Server /http *# set http-redirect yes
Server /http *# set timeout 1800
Server /http *# commit
Server /http # show
HTTP Port  HTTPS Port Timeout  Active Sessions HTTPS Enabled HTTP Redirected  HT
                                  TP Enabled
---------- ---------- -------- --------------- ------- ---------------- --------
                                  --------
80         443        1800     0               yes     yes              yes
Server /http #
```

Configuring SSH

Before you begin

You must log in as a user with admin privileges to configure SSH.

SUMMARY STEPS

1. Server# **scope ssh**
2. Server /ssh # **set enabled** {**yes** | **no**}
3. Server /ssh # **set ssh-port** *number*
4. Server /ssh # **set timeout** *seconds*
5. Server /ssh # **commit**
6. Server /ssh # **show** [**detail**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	Server# **scope ssh**	Enters the SSH command mode.
Step 2	Server /ssh # **set enabled** {**yes** | **no**}	Enables or disables SSH on the .
Step 3	Server /ssh # **set ssh-port** *number*	Sets the port to use for secure shell access. The default is 22.
Step 4	Server /ssh # **set timeout** *seconds*	Sets the number of seconds to wait before the system considers an SSH request to have timed out. Enter an integer between 60 and 10,800. The default is 300 seconds.
Step 5	Server /ssh # **commit**	Commits the transaction to the system configuration.
Step 6	Server /ssh # **show** [**detail**]	(Optional) Displays the SSH configuration.

Example

This example configures SSH for the :

```
Server# scope ssh
Server /ssh # set enabled yes
Server /ssh *# set ssh-port 22
Server /ssh *# set timeout 600
Server /ssh *# commit
Server /ssh # show
SSH Port   Timeout  Active Sessions Enabled
---------- -------- --------------- -------
22         600      1               yes

Server /ssh #
```

Configuring XML API

XML API for

The Cisco XML application programming interface (API) is a programmatic interface to for a C-Series Rack-Mount Server. The API accepts XML documents through HTTP or HTTPS.

For detailed information about the XML API, see *Cisco UCS Rack-Mount Servers XML API Programmer's Guide*.

Enabling XML API

Before you begin

You must log in as a user with admin privileges to perform this task.

SUMMARY STEPS

1. Server# **scope xmlapi**
2. Server /xmlapi # **set enabled** {**yes** | **no**}
3. Server /xmlapi # **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	Server# **scope xmlapi**	Enters XML API command mode.
Step 2	Server /xmlapi # **set enabled** {**yes** | **no**}	Enables or disables XML API control of .
Step 3	Server /xmlapi # **commit**	Commits the transaction to the system configuration.

Example

This example enables XML API control of and commits the transaction:

```
Server# scope xmlapi
Server /xmlapi # set enabled yes
Server /xmlapi *# commit
Server /xmlapi #  show detail
XMLAPI Settings:
    Enabled: yes
    Active Sessions: 0
    Max Sessions: 4

Server /xmlapi #
```

Enabling Redfish

Before you begin

You must log in as a user with admin privileges to perform this task.

SUMMARY STEPS

1. Server# **scope redfish**
2. Server /redfish # **set enabled** {**yes** | **no**}
3. Server /redfish* # **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	Server# **scope redfish**	Enters redfish command mode.
Step 2	Server /redfish # **set enabled** {**yes**	**no**}	Enables or disables redfish control of .
Step 3	Server /redfish* # **commit**	Commits the transaction to the system configuration.

Example

This example enables redfish control of and commits the transaction:

```
Server# scope redfish
Server /redfish # set enabled yes
Server /redfish *# commit
Server /redfish #  show detail
REDFISH Settings:
    Enabled: yes
    Active Sessions: 0
    Max Sessions: 4

Server /redfish #
```

Configuring IPMI

IPMI Over LAN

Intelligent Platform Management Interface (IPMI) defines the protocols for interfacing with a service processor embedded in a server platform. This service processor is called a Baseboard Management Controller (BMC) and resides on the server motherboard. The BMC links to a main processor and other on-board elements using a simple serial bus.

During normal operations, IPMI lets a server operating system obtain information about system health and control system hardware. For example, IPMI enables the monitoring of sensors, such as temperature, fan speeds and voltages, for proactive problem detection. If server temperature rises above specified levels, the

server operating system can direct the BMC to increase fan speed or reduce processor speed to address the problem.

Configuring IPMI over LAN for Cisco IMC

Configure IPMI over LAN when you want to manage the with IPMI messages.

Before you begin

You must log in with admin privileges to perform this task.

SUMMARY STEPS

1. Server # **scope server** {**1** | **2**}
2. Server /server # **scope ipmi**
3. Server /server/ipmi # **set enabled** {**yes** | **no**}
4. Server /server/ipmi # **set privilege-level** {**readonly** | **user** | **admin**}
5. Server /server/ipmi # **set encryption-key** *key*
6. Server /server/ipmi # **commit**
7. Server /server/ipmi # **randomise-key**
8. At the prompt, enter **y** to randomize the encryption key.

DETAILED STEPS

	Command or Action	Purpose
Step 1	Server # **scope server** {**1** | **2**}	Enters server command mode of server 1 or 2.
Step 2	Server /server # **scope ipmi**	Enters the IPMI command mode.
Step 3	Server /server/ipmi # **set enabled** {**yes** | **no**}	Enables or disables IPMI access on this server.
Step 4	Server /server/ipmi # **set privilege-level** {**readonly** | **user** | **admin**}	Specifies the highest privilege level that can be assigned to an IPMI session on this server. This can be: • **readonly** — IPMI users can view information but cannot make any changes. If you select this option, IPMI users with the "Administrator", "Operator", or "User" user roles can only create read-only IPMI sessions, regardless of their other IPMI privileges. • **user** — IPMI users can perform some functions but cannot perform administrative tasks. If you select this option, IPMI users with the "Administrator" or "Operator" user role can create user and read-only sessions on this server. • **admin** — IPMI users can perform all available actions. If you select this option, IPMI users with the "Administrator" user role can create admin, user, and read-only sessions on this server.

	Command or Action	Purpose
Step 5	Server /server/ipmi # **set encryption-key** *key*	Sets the IPMI encryption key to use for IPMI communications. The key value must be 40 hexadecimal numbers.
Step 6	Server /server/ipmi # **commit**	Commits the transaction to the system configuration.
Step 7	Server /server/ipmi # **randomise-key**	Sets the IPMI encryption key to a random value. **Note** You can perform the Step 6 action instead of Steps 4 and 5.
Step 8	At the prompt, enter **y** to randomize the encryption key.	Sets the IPMI encryption key to a random value.

Example

This example configures IPMI over LAN for the :

```
Server # scope server 1
Server /server # scope ipmi
Server /server/ipmi # set enabled yes
Server /server/ipmi *# set privilege-level admin
Server /server/ipmi *# set encryption-key abcdef01234567890abcdef01234567890abcdef
Server /server/ipmi *# commit
Server /server/ipmi *# show
Enabled Encryption Key                          Privilege Level Limit
------- --------------------------------------- ---------------------
yes     ABCDEF01234567890ABCDEF01234567890ABCDEF admin

Server /server/ipmi # randomise-key
This operation will change the IPMI Encryption Key to a random value
Continue?[y|N]y
Setting IPMI Encryption Key to a random value...

Server /server/ipmi # show
Enabled Encryption Key                          Privilege Level Limit
------- --------------------------------------- ---------------------
yes     abcdef01234567890abcdef01234567890abcdef admin

Server /server/ipmi #
```

Configuring IPMI over LAN for CMCs

Configure IPMI over LAN when you want to manage the CMC with IPMI messages.

Before you begin

You must log in with admin privileges to perform this task.

SUMMARY STEPS

1. Server # **scope chassis**
2. Server /chassis # **scope cmc** {**1** | **2**}
3. Server /server # **scope ipmi**

4. Server /chassis/cmc/ipmi # **set enabled** {**yes** | **no**}
5. Server /chassis/cmc/ipmi # **set privilege-level** {**readonly** | **user** | **admin**}
6. Server /chassis/cmc/ipmi # **set encryption-key** *key*
7. Server /chassis/cmc/ipmi # **commit**
8. Server /chassis/cmc/ipmi # **randomise-key**
9. At the prompt, enter **y** to randomize the encryption key.

DETAILED STEPS

	Command or Action	Purpose
Step 1	Server # **scope chassis**	Enters server command mode of server 1 or 2.
Step 2	Server /chassis # **scope cmc** {**1** | **2**}	Enters CMC command mode.
Step 3	Server /server # **scope ipmi**	Enters the IPMI command mode.
Step 4	Server /chassis/cmc/ipmi # **set enabled** {**yes** | **no**}	Enables or disables IPMI access on this server.
Step 5	Server /chassis/cmc/ipmi # **set privilege-level** {**readonly** | **user** | **admin**}	Specifies the highest privilege level that can be assigned to an IPMI session on this server. This can be: • **readonly** — IPMI users can view information but cannot make any changes. If you select this option, IPMI users with the "Administrator", "Operator", or "User" user roles can only create read-only IPMI sessions, regardless of their other IPMI privileges. • **user** — IPMI users can perform some functions but cannot perform administrative tasks. If you select this option, IPMI users with the "Administrator" or "Operator" user role can create user and read-only sessions on this server. • **admin** — IPMI users can perform all available actions. If you select this option, IPMI users with the "Administrator" user role can create admin, user, and read-only sessions on this server.
Step 6	Server /chassis/cmc/ipmi # **set encryption-key** *key*	Sets the IPMI encryption key to use for IPMI communications. The key value must be 40 hexadecimal numbers.
Step 7	Server /chassis/cmc/ipmi # **commit**	Commits the transaction to the system configuration.
Step 8	Server /chassis/cmc/ipmi # **randomise-key**	Sets the IPMI encryption key to a random value. **Note** You can perform the Step 6 action instead of Steps 4 and 5.
Step 9	At the prompt, enter **y** to randomize the encryption key.	Sets the IPMI encryption key to a random value.

Example

This example configures IPMI over LAN for the CMC 1:

```
Server # scope chassis
Server # scope cmc 1
Server /chassis # scope ipmi
Server /chassis/cmc/ipmi # set enabled yes
Server /chassis/cmc/ipmi *# set privilege-level admin
Server /chassis/cmc/ipmi *# set encryption-key abcdef01234567890abcdef01234567890abcdef
Server /chassis/cmcipmi *# commit
Server /chassis/cmc/ipmi *# show
Enabled Encryption Key                          Privilege Level Limit
------- --------------------------------------- ---------------------
yes     ABCDEF01234567890ABCDEF01234567890ABCDEF admin

Server /chassis/cmc/ipmi # randomise-key
This operation will change the IPMI Encryption Key to a random value
Continue?[y|N]y
Setting IPMI Encryption Key to a random value...

Server /chassis/cmc/ipmi # show
Enabled Encryption Key                          Privilege Level Limit
------- --------------------------------------- ---------------------
yes     abcdef01234567890abcdef01234567890abcdef admin

Server /chassis/cmc/ipmi #
```

Configuring SNMP

SNMP

The Cisco UCS C-Series Rack-Mount Servers support the Simple Network Management Protocol (SNMP) for viewing server configuration and status and for sending fault and alert information by SNMP traps. For information on Management Information Base (MIB) files supported by Cisco IMC, see the *MIB Quick Reference for Cisco UCS* at this URL: http://www.cisco.com/c/en/us/td/docs/unified_computing/ucs/sw/mib/b-series/b_UCS_MIBRef.html.

Beginning with release 4.1(3b), Cisco IMC introduces enhanced authentication protocol for SNMP v3 version. SNMP v3 users cannot be added with **DES** security protocol.

Cisco IMC GUI displays a warning when you select an existing v3 version with unsupported security level, authentication type, or privacy type. You may select and modify the user details.

Configuring SNMP Properties

This procedure is applicable for Cisco UCS C-Series M6 and earlier servers. To configure SNMP user for Cisco UCS C-Series M7 and later servers, see Configuring Local Users for Cisco UCS C-Series M7 and Later Servers.

Before you begin

You must log in as a user with admin privileges to perform this task.

Procedure

	Command or Action	Purpose
Step 1	Server# **scope snmp**	Enters SNMP command mode.
Step 2	Server /snmp # **set enabled** {**yes** | **no**}	Enables or disables SNMP. **Note** SNMP must be enabled and saved before additional SNMP configuration commands are accepted.
Step 3	Server /snmp # **commit**	Commits the transaction to the system configuration.
Step 4	Server /snmp # **set enable-serial-num** {**yes** | **no**}	Prefixes the traps with the serial number of the server.
Step 5	Server /snmp # **set community-str** *community*	Specifies the default SNMP v1 or v2c community name that includes on any trap messages it sends to the SNMP host. The name can be up to 18 characters.
Step 6	Server /snmp # **set community-access**	This can be one of the following : Disabled, Limited, or Full.
Step 7	Server /snmp # **set trap-community-str**	Specifies the SNMP community group to which trap information should be sent. The name can be up to 18 characters
Step 8	Server /snmp # **set sys-contact** *contact*	Specifies the system contact person responsible for the SNMP implementation. The contact information can be up to 254 characters, such as an email address or a name and telephone number. To enter a value that contains spaces, you must enclose the entry with quotation marks.
Step 9	Server /snmp # **set sys-location** *location*	Specifies the location of the host on which the SNMP agent (server) runs. The location information can be up to 254 characters. To enter a value that contains spaces, you must enclose the entry with quotation marks.
Step 10	Server /snmp # **commit**	Commits the transaction to the system configuration.

Example

This example configures the SNMP properties and commits the transaction:

What to do next

Configure SNMP trap settings as described in Configuring SNMP Trap Settings, on page 14.

Configuring SNMP Trap Settings

Before you begin

- You must log in with admin privileges to perform this task.
- SNMP must be enabled and saved before trap settings can be configured.

Procedure

	Command or Action	Purpose
Step 1	Server# **scope snmp**	Enters the SNMP command mode.
Step 2	Server /snmp # **scope trap-destinations** *number*	Enters the SNMP trap destination command mode for the specified destination. Four SNMP trap destinations are available. The destination *number* is an integer between 1 and 15.
Step 3	Server /snmp/trap-destinations # **set enabled** {**yes** | **no**}	Enables or disables the SNMP trap destination.
Step 4	Server /snmp/trap-destinations # **set version** { | **2** | **3**}	Specify the desired SNMP version of the trap message. **Note** SNMPv3 traps will be delivered only to locations where the SNMPv3 user and key values are configured correctly.
Step 5	Server /snmp/trap-destinations # **set type** {**trap** | **inform**}	Specifies whether SNMP notification messages are sent as simple traps or as inform requests requiring acknowledgment by the receiver. **Note** The inform option can be chosen only for V2 users.
Step 6	Server /snmp/trap-destinations # **set user** *user*	**Note** While Configuring SNMP v3 version, you cannot use SNMP users with Encryption Method set as **DES**.
Step 7	Server /snmp/trap-destination # **commit**	Commits the transaction to the system configuration.

Example

This example configures general SNMP trap settings and trap destination number 1 and commits the transaction:

```
Server# scope snmp
Server /snmp # Scope trap-destinations 1
Server /snmp/trap-destination *# set enabled yes
Server /snmp/trap-destination *#  set version 2
Server /snmp/trap-destination *# set type inform
Server /snmp/trap-destination *# set user user1
```

```
Server /snmp/trap-destination *# commit
Server /snmp/trap-destination # show detail
Trap Destination 1:
    Enabled: yes
    SNMP version: 2
    Trap type: inform
    SNMP user: user1


    Delete Trap: no
Server /snmp/trap-destination #
```

Sending a Test SNMP Trap Message

Before you begin

You must log in with admin privileges to perform this task.

Procedure

	Command or Action	Purpose
Step 1	Server# **scope snmp**	Enters the SNMP command mode.
Step 2	Server /snmp # **send-test-trap**	Sends an SNMP test trap to the configured SNMP trap destination that are enabled. **Note** The trap must be configured and enabled in order to send a test message.

Example

This example sends a test message to all the enabled SNMP trap destinations:

```
Server# scope snmp
Server /snmp #  send-test-trap
SNMP Test Trap sent to the destination.
Server /snmp #
```

Configuring SNMPv3 Users

Before you begin

- You must log in as a user with admin privileges to perform this task.
- SNMP must be enabled and saved before these configuration commands are accepted.

SUMMARY STEPS

1. Server# **scope snmp**
2. Server /snmp # **scope v3users** *number*

3. Server /snmp/v3users # **set v3add** {**yes** | **no**}
4. Server /snmp/v3users # **set v3security-name** *security-name*
5. Server /snmp/v3users # **set v3security-level** {**noauthnopriv** | **authnopriv** | **authpriv**}
6. Server /snmp/v3users # **set v3proto** {**MD5** | **SHA**}
7. Server /snmp/v3users # **set v3auth-key** *auth-key*
8. Server /snmp/v3users # **set v3priv-proto** {**DES** | **AES**}
9. Server /snmp/v3users # **set v3priv-auth-key** *priv-auth-key*
10. Server /snmp/v3users # **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	Server# **scope snmp**	Enters the SNMP command mode.
Step 2	Server /snmp # **scope v3users** *number*	Enters the SNMPv3 users command mode for the specified user number.
Step 3	Server /snmp/v3users # **set v3add** {**yes** | **no**}	Adds or deletes an SNMPv3 user. This can be one of the following: • **yes**—This user is enabled as an SNMPv3 user and is allowed to access the SNMP OID tree. **Note** The security name and security level must also be configured at this time or the user addition will fail. • **no**—This user configuration is deleted.
Step 4	Server /snmp/v3users # **set v3security-name** *security-name*	Enter an SNMP username for this user.
Step 5	Server /snmp/v3users # **set v3security-level** {**noauthnopriv** | **authnopriv** | **authpriv**}	Select a security level for this user. This can be one of the following: • **noauthnopriv**—The user does not require an authorization or privacy password. • **authnopriv**—The user requires an authorization password but not a privacy password. If you select this option, you must configure an authentication key. • **authpriv**—The user requires both an authorization password and a privacy password. If you select this option, you must configure an authentication key and a private encryption key. **Note** For a v3 version, only authnopriv and authpriv security levels are available.
Step 6	Server /snmp/v3users # **set v3proto** {**MD5** | **SHA**}	**Note** For a v3 version, only SHA authentication methods are available.

	Command or Action	Purpose
		Select an authentication protocol for this user.
Step 7	Server /snmp/v3users # **set v3auth-key** *auth-key*	Enter an authorization password for this user.
Step 8	Server /snmp/v3users # **set v3priv-proto** {**DES** | **AES**}	**Note** For a v3 version, only AES option is available. Select an encryption protocol for this user.
Step 9	Server /snmp/v3users # **set v3priv-auth-key** *priv-auth-key*	Enter a private encryption key (privacy password) for this user.
Step 10	Server /snmp/v3users # **commit**	Commits the transaction to the system configuration.

Example

This example configures SNMPv3 user number 2 and commits the transaction:

```
Server# scope snmp
Server /snmp # scope v3users 2
Server /snmp/v3users # set v3add yes
Server /snmp/v3users *# set v3security-name ucsSNMPV3user
Server /snmp/v3users *# set v3security-level authpriv
Server /snmp/v3users *# set v3proto SHA
Server /snmp/v3users *# set v3auth-key
Please enter v3auth-key:ex4mp1ek3y
Please confirm v3auth-key:ex4mp1ek3y
Server /snmp/v3users *# set v3priv-proto AES
Server /snmp/v3users *# set v3priv-auth-key
Please enter v3priv-auth-key:!1@2#3$4%5^6&7*8
Please confirm v3priv-auth-key:!1@2#3$4%5^6&7*8
Server /snmp/v3users *# commit
Settings are being applied ... allow a few minutes for the process to complete
Server /snmp/v3users # show detail
User 2:
    Add User: yes
    Security Name: ucsSNMPV3user
    Security Level: authpriv
    Auth Type: SHA
    Auth Key: ******
    Encryption: AES
    Private Key: ******

Server /snmp/v3users #
```

Configuring a Server to Send Email Alerts Using SMTP

The Cisco IMC supports email-based notification of server faults to recipients without relying on the SNMP. The system uses the Simple Mail Transfer Protocol (SMTP) to send server faults as email alerts to the configured SMTP server.

A maximum of four recipients is supported.

Configuring SMTP Servers for Receiving E-Mail Alerts

Before you begin

You must log in as a user with admin privileges to perform this task.

SUMMARY STEPS

1. Server# **scope smtp**
2. Server /smtp # **set enabled** {**yes** | **no**}
3. Server /smtp * # **set server-addr** *IP_Address*
4. Server /smtp * # **set port** *port_number*
5. Server /smtp # **set-mail-addr** *email_address recipient_minimum_severity informational* | *warning* | *minor* | *major* | *critical*
6. Server /smtp * # **commit**
7. Server /smtp # **send-test-mail recipient1**

DETAILED STEPS

	Command or Action	Purpose
Step 1	Server# **scope smtp**	Enters the SMTP command mode.
Step 2	Server /smtp # **set enabled** {**yes** | **no**}	Enables or disables the SMTP feature.
Step 3	Server /smtp * # **set server-addr** *IP_Address*	Assigns the SMTP server IP address.
Step 4	Server /smtp * # **set port** *port_number*	Sets the port number for the SMTP server.
Step 5	Server /smtp # **set-mail-addr** *email_address recipient_minimum_severity informational* | *warning* | *minor* | *major* | *critical*	Sets recipient email address with minimum severity level.
Step 6	Server /smtp * # **commit**	Commits the transaction to the system configuration.
Step 7	Server /smtp # **send-test-mail recipient1**	Sends a test mail alert to the email address assigned to the chosen recipient.

Example

This example shows how to configure SMTP for receiving mail alerts:

```
Server # scope smtp
Server /smtp # set enabled yes
Server /smtp *# set server-addr 10.10.10.10
Server /smtp *# set port 25
Server /smtp *# set-mail-addr recipient4 user@cisco.com critical
This operation will add the recipient4
Continue?[y|N]y
Server /smtp *#
Server /smtp *# commit
Server /smtp #
```