Configuring Authentication, Authorization, and Accounting

Last Updated: December 2, 2010

This chapter contains procedures for:

Overview

Cisco Unified Messaging Gateway supports Authentication, Authorization, and Accounting (AAA) which enables you to determine which users could access restricted services by assigning predefined privileges to groups.

You can create privileges and then assign these privileges to groups. Cisco UMG supports the following features:

- The ability to log AAA accounting information that enables you to easily audit configuration changes, maintain security, accurately allocate resources, and determine who should be billed for the use of resources.
- The ability to use a remote RADIUS server for authentication.
- The ability to configure failover capabilities to for the accounting and authentication servers.

To configure the AAA features, use the following procedures:

Configuring the Accounting Server

You can configure up to two AAA accounting servers. Automatic failover functionality is provided if you have two accounting servers configured. In this case, if the first server is unreachable, the accounting information is sent the second server. If both accounting servers are unreachable, accounting records are cached until a server becomes available. If a server cannot be reached before the cache is full, the oldest accounting packets are dropped to make room for the new packets.

Because the configuration of the AAA accounting server is completely independent of the AAA authentication server, you can configure the AAA accounting server to be on the same or different machine from the AAA authentication server.

If you use a syslog server, it is not affected by the AAA configuration and continues to use the existing user interfaces. When the RADIUS server sends AAA accounting information to a syslog server, it is normalized into a single string before being recorded. If no syslog server is defined, the AAA accounting logs are recorded by the syslog server running locally on Cisco UMG.

For an accounting server, you can configure the following information used to log into the server:

- Server IP address or DNS name
- Port number used
- Cryptographic shared secret and security credentials
- Number of login retries
- Length of login timeout

Note Only RADIUS servers are supported.

Specifying AAA Accounting Settings

SUMMARY STEPS

1. **config t**
2. **aaa accounting server remote**
3. **address** *address* **[port** *port***] secret** *secret*
4. **address** *address* **[port** *port***] credentials hidden** *cred*
5. **retries** *number*
6. **timeout** *seconds*
7. **end**
8. **show aaa accounting service**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**config t** **Example:** `umg-1# config t`	Enters configuration mode.
Step 2	**aaa accounting server remote** **Example:** `umg-1(config)# aaa accounting server remote`	Enters aaa-authentication submode to enable you to configure the AAA authentication server.
Step 3	**address** *address* [**port** *port*] **secret** *secret* **Example:** `umg-1(config)# address 10.2.2.10 prt 1808 secret ezsecret`	Defines the access parameters for the AAA accounting server.
Step 4	**address** *address* [**port** *port*] **credentials hidden** *cred* **Example:**	Defines the access parameters for the AAA accounting server.
Step 5	**retries** *number* **Example:** `umg-1(config)# retries 6`	Specifies the maximum number of times an AAA accounting request is retried before the accounting request fails.
Step 6	**timeout** *seconds* **Example:** `umg-1(config)# timeout 24`	Specifies the amount of time to wait before an AAA accounting request is considered to be unanswered.
Step 7	**end** **Example:** `umg-1(config)# end`	Exits to privileged EXEC mode.
Step 8	**show aaa accounting service** **Example:** `umg-1# show aaa accounting service`	(Optional) Displays the settings for the AAA accounting server.

Examples

The following is sample output from the **show aaa accounting service** command:

```
umg-1# show aaa accounting service
AAA Accounting Service Configuration
Accounting: Enabled
Address: 192.168.1.101 Port: 1813 Credentials:
EugxIjn3MbL3WgUZUdUb90nfGWTYHfmPSd8ZZNgd+Y9J3xlk2B35j0nfGWTYHfmPSd8ZZNgd+Y9J3xlk2B35j0nfGW
TYHfmPSd8ZZNgd+Y9J3xlk2B35j0nfGWTYHfmP
Address: 192.168.1.100 Port: 1813 Credentials:
EugxIjn3MbL3WgUZUdUb90nfGWTYHfmPSd8ZZNgd+Y9J3xlk2B35j0nfGWTYHfmPSd8ZZNgd+Y9J3xlk2B35j0nfGW
TYHfmPSd8ZZNgd+Y9J3xlk2B35j0nfGWTYHfmP
Timeout: 5 (sec)
Retries: 3
```

Configuring the Authentication Server

The two procedures for configuring AAA authentication consist of:

- Configuring connection parameters for the AAA authentication server
- Configuring whether the authentication servers or local authentication database will be queried first

This section covers only the first procedure. The second procedure is covered in the "Configuring the AAA Policy" section on page 114.

For an AAA authentication server, you can configure the following information used to log into the server:

- Server IP address or DNS name
- Port number used
- Cryptographic shared secret and security credentials
- Number of login retries
- Length of login timeout

Note To help protect the cryptographic information of the RADIUS server, you must view the running configuration to see this information.

Specifying AAA Authentication Settings

SUMMARY STEPS

1. **config t**
2. **aaa authentication server remote**
3. **address** *address* [**port** *port*] **secret** *secret*
4. **address** *address* [**port** *port*] **credentials hidden** *cred*
5. **retries** *number*
6. **timeout** *seconds*

7. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**config t** **Example:** `umg-1# config t`	Enters configuration mode.
Step 2	**aaa authentication server remote** **Example:** `umg-1(config)# aaa authentication server remote`	Enters aaa-authentication submode to enable you to configure the AAA authentication server.
Step 3	**address** *address* [**port** *port*] **secret** *secret* **Example:** `umg-1(config)# address 10.2.2.10 port 1808 secret ezsecret`	Defines the access parameters for the AAA authentication server.
Step 4	**address** *address* [**port** *port*] **credentials hidden** *cred* **Example:**	Defines the access parameters for the AAA authentication server.
Step 5	**retries** *number* **Example:** `umg-1(config)# retries 6`	Specifies maximum number of times an AAA authentication request is retried before the authentication request fails.
Step 6	**timeout** *seconds* **Example:** `umg-1(config)# timeout 24`	Specifies the amount of time to wait before an AAA authentication request is considered unanswered.
Step 7	**end** **Example:** `umg-1(config)# end`	Exits to privileged EXEC mode.

Configuring the AAA Policy

The AAA policy specifies the failover functionality that you can optionally configure for the authentication server. You can choose from two types of failover functionality:

- Authentication failover
- Unreachable failover

You can also use a combination of both failover methods.

Authentication Failover

The authentication failover feature enables you to optionally use a remote RADIUS server for user login authentication in addition to the local database. The procedure in this section configures the order in which authentication is resolved. You can configure authentication to use:

- Only the local database
- Only the remote server
- The local database first, then the remote server
- The remote server first, then the local database

Note The authentication failover feature has the following limitations:

- Authentication with a RADIUS server is available only when accessing the GUI or CLI interface and requires only a user ID and password.
- Login information is not synchronized between the local system and the remote server. Any security features such, as password expiration, must be configured separately for Cisco Unified Messaging Gateway and the RADIUS server.

Unreachable Failover

The unreachable failover is used only with RADIUS servers. This feature enables you to configure up to two addresses that can be used to access RADIUS servers.

As Cisco UMG attempts to authenticate a user with the RADIUS servers, messages are sent to users to notify them when a RADIUS server:

- Cannot be reached
- Fails to authenticate the user

Example

In this example, authentication is performed by the remote server first, then by the local database. Also, two addresses are configured for the remote RADIUS server.

This is a sequence of events that could occur during authentication for this example:

1. Cisco UMG tries to contact the first remote RADIUS server.

2. If the first RADIUS server does not respond or does not accept the authentication credentials of the user, Cisco UMG tries to contact the second remote RADIUS server.
3. If the second RADIUS server does not respond or does not accept the authentication credentials of the user, the user receives the appropriate error message and Cisco UMG tries to contact the local database.
4. If the local database does not accept the authentication credentials of the user, the user receives an error message.

Specifying the Policy that Controls the Behavior of Authentication and Authorization

SUMMARY STEPS

1. **config t**
2. **aaa policy system**
3. **authentication-order** {**remote** [**local**] | **local** [**remote**]}
4. **authorization merge-attributes**
5. **end**
6. **show aaa policy**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**config t** **Example:** `umg-1# config t`	Enters configuration mode.
Step 2	**aaa policy system** **Example:** `umg-1(config)# aaa policy system`	Enters aaa-authentication submode to enable you to specify the policy that controls the behavior of authentication and authorization.
Step 3	**authentication-order** {**remote** [**local**]|**local** [**remote**]} **Example:** `umg-1(config)# authentication-order remote local`	Specifies the order in which to query the authentication servers and local authentication database.
Step 4	**authorization merge-attributes** **Example:** `umg-1(config)# authorization merge-attributes`	Specifies whether the user attributes that are retrieved from a remote RADIUS AAA server are merged with attributes for the same username found in the local user database.

	Command or Action	Purpose
Step 5	**end** **Example:** `umg-1(config)# end`	Exits to privileged EXEC mode.
Step 6	**show aaa accounting policy** **Example:** `umg-1# show aaa policy`	(Optional) Displays the AAA policy settings.

Examples

The following is sample output from the **show aaa policy** command:

```
umg-1# show aaa policy
authentication-order local
merge-attributes enable
preferred-server remote
```

Configuring Privileges

Cisco UMG software provides several predefined privileges that you can assign to groups. You can also create your own privileges and modify the predefined privileges.

When you assign a privilege to a group, any member of the group is granted the privilege rights. An administrator group is created automatically by the software initialization process from the imported subscribers designated as administrators.

When you create or modify privileges, you add or delete the operations allowed by that privilege. Operations define the CLI commands and GUI functions that are allowed. In addition to adding operations to a privilege, you can also configure a privilege to have another privilege nested inside of it. A privilege configured with a nested privilege includes all operations configured for the nested privilege.

As part of the planning process, you should decide:

- How many categories of user privileges you want to create for your company.
- Which functions each privilege will allow your users to perform.

After you decide which privileges you want your users to have:

1. Review the predefined privileges to determine whether any of them are similar to the permissions that you want to give to each of your categories of users.
2. Configure a separate privilege for each category by specifying which operations each category of users will be allowed to preform, optionally including predefined privileges (see "Creating and Customizing Privileges" on page 121).
3. Create a group for each category of user privilege and assign the appropriate privilege to each group of users.
4. Add your users to the appropriate group.

Tip For an example of the commands used for these steps, see the "Configuration Example" section on page 119.

Note You cannot modify the superuser privilege.

Table 3 describes the predefined privileges provided with the Cisco UMG software and the operations associated with them. Table 4 describes all available operations that you can add to privileges.

To display a list of privileges, use the **show privileges** command in Cisco UMG EXEC mode. To display detailed information about a specific privilege, use the **show privilege detail** command.

Table 3 Privileges

Privilege	Description	Operations
Superuser	Grants unrestricted system access.	all
Broadcast	Allows subscribers to send broadcast messages across the network.	broadcast.local, broadcast.remote, system.debug
Local-broadcast	Allows subscribers to send broadcast messages only to subscribers on the local network.	broadcast.local, system.debug
ViewRealTime Reports	Allows subscribers to view real-time reports	report.realtime
manage-users	Allows subscribers to create, modify, and delete users	user.configuration, user.pin, user.password, user.mailbox, user.notification, user.remote, group.configuration, system.debug
manage-passwords	Allows subscribes to create, modify, and delete user passwords and PINs	user.pin, user.password, system. debug

Table 4 List of Operations

Operation	Description
group.configuration	Create, modify, and delete groups.
security.aaa	Configure and modify AAA service settings.
security.access	Configure system level security regarding encryption of data, including defining crypto keys. **Note** Also includes permission to reload the system.

Table 4 List of Operations (continued)

Operation	Description
security.password	Configure settings for the system password and policy, such as: • Expiry • Lockout (temporary and permanent) • History • Length
security.pin	Configure settings for the system PIN and policy, such as: • Expiry • Lockout (temporary and permanent) • History • Length
services.configuration	Configure system services: DNS, NTP/clock, SMTP, SNMP, Fax Gateway, Cisco UMG, hostname, domain, interfaces (counters), and system default language. **Note** Also includes permission to reload the system.
services.manage	System level services commands not related to configuration like clearing DNS cache and ping.
software.install	Install, upgrade, or inspect system software or add-ons such as languages and licenses. **Note** Also includes permission to reload the system.
system.backup	Configure backup.
system.configuration	Configure system settings such as the clock, hostname, domain name, default language, and interfaces (counters).
system.debug	Collect and configure trace and debug data. Includes copying data like core and log files.
system.view	View system settings and configuration.
user.configuration	Create, modify, and delete users and groups, including the configuration of: • First and Last Name • Nickname • Display Name • Language
user.password	Create, set, or remove others passwords.
user.pin	Create, set, or remove others PINs.

Configuration Example

In this example, a company wants a security structure with two levels of security administration. The two levels allow the following actions to be taken by the administrator:

- The first level enables the security administrator to reset the passwords and PINs for users that have locked themselves out of the system, whether they forgot their password or their account is locked because of too many failed login attempts. This level will be called PASSWORD RESET.
- The second level enables the security administrator to act as a system guardian by:
  - Ensuring that the proper security policies are implemented for issues such as password aging, account lockout, encryption, authentication, authorization, and accounting
  - Ensuring that data remain safe from attackers without over burdening end users with security related details and tasks
  - Monitoring the system to ensure that only legitimate users have access
  - Troubleshooting any problems that legitimate users have with accessing the system
  - Resetting passwords and PINs for users that have locked themselves out of the system, whether they forgot their password or their account is locked because of too many failed login attempts

  This level is called SYSTEM GUARDIAN.

When you use the general planning and configuration steps as described in the "Configuring Privileges" section on page 116, to set up the security administration levels for this example, these are the results:

- You have already decided:
  - How many levels or categories of user privileges you want to create for your company
  - Which functions each privilege will allow your users to perform

  There will be two levels, called PASSWORD RESET and SYSTEM GUARDIAN, as described above.
- After reviewing the predefined privileges to determine whether any of them are similar to the permissions that you want to give each of your security levels, you find that:
  - The predefined privilege called *manage-passwords* can be used for the security level named PASSWORD RESET because it has all of the permissions needed to help users that have locked themselves out of the system.
  - The *manage-passwords* privilege also has a subset of the permissions needed the security level named SYSTEM GUARDIAN and is the predefined privilege closest to your requirements. However, to act as system guardian, the following additional operations will have to included: *security.access*, *security.aaa*, *security.password*, *security.pin*, *system.debug*, and *system.view*. See Table 4 on page 117 for more information.
- Use the following commands to configure a privilege for the SYSTEM GUARDIAN security level by including the predefined privilege *manage-password* and adding the operations listed in the previous bullet:

```
umg-1(config)# privilege guardian-privilege create
umg-1(config)# privilege guardian-privilege member manage-passwords
umg-1(config)# privilege guardian-privilege operation security.access
umg-1(config)# privilege guardian-privilege operation security.aaa
umg-1(config)# privilege guardian-privilege operation security.password
umg-1(config)# privilege guardian-privilege operation security.pin
umg-1(config)# privilege guardian-privilege operation system.debug
umg-1(config)# privilege guardian-privilege operation system.view
```

Note You do not have to configure a privilege for the PASSWORD RESET security level because you can use the predefined privilege *manage-passwords*.

- Use the following commands to create a new group called *password-reset* and assign the privilege called *manage-password*s to it:

```
umg-1(config)# groupname password-reset create
umg-1(config)# groupname password-reset privilege manage-passwords
```

- Use the following commands to create a new group called *system-guardian* and assign the privilege called *guardian-privilege*:

```
umg-1(config)# groupname system-guardian create
umg-1(config)# groupname system-guardian privilege guardian-privilege
```

- Assign the appropriate users to the new groups, associating them with their roles. For example, if you want Bob and Ned to have the privileges of the PASSWORD RESET security administration level and Ann to have the privileges of the SYSTEM GUARDIAN security administration level, use the following commands:

```
umg-1(config)# groupname password-reset member bob
umg-1(config)# groupname password-reset member ned
umg-1(config)# groupname system-guardian member ann
```

- The configuration of this example is now complete. You can verify your configuration using the following commands.

  The following is sample output from the **show group detail groupname password-reset expanded** command:

```
umg-1# show group detail groupname password-reset expanded
Groupname:          password-reset
Full Name:          password-reset
Description:
Email:
Epage:

Group Members:      <none>
User Members:       bob ned
Group Owners:       <none>
User Owners:        <none>
Privileges:         manage-passwords
```

  The following is sample output from the **show group detail groupname system-guardian expanded** command:

```
umg-1# show group detail groupname system-guardian expanded
Groupname:          system-guardian
Full Name:          system-guardian
Description:
Email:
Epage:

Group Members:      <none>
User Members:       ann
Group Owners:       <none>
User Owners:        <none>
Privileges:         guardian-privilege
```

  The following is sample output from the **show privilege detail manage-passwords expanded** command:

```
umg-1# show privilege detail manage-passwords expanded
Privilege:          manage-passwords
Description:        Privilege to reset user passwords

Privilege Members:  <none>
Operations:         system.debug user.password user.pin
```

The following is sample output from the **show privilege detail guardian-privilege expanded** command:

```
umg-1# show privilege detail guardian-privilege expanded
Privilege:          guardian-privilege
Description:

Privilege Members:  manage-passwords
Operations:         security.aaa security.access security.password security.pin
                    system.debug system.view
   manage-passwords:system.debug user.password user.pin
```

Creating and Customizing Privileges

SUMMARY STEP

1. **config t**
2. **privilege** *privilege-name* **create**
3. **privilege** *privilege-name* **description** *string*
4. **privilege** *privilege-name* **operation** *operation-name*
5. **privilege** *privilege-name* **member** *privilege-name2*
6. **end**
7. **show operations**
8. **show operation detail** *operation-name*
9. **show privileges**
10. **show privilege detail** *privilege-name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**config t** **Example:** `umg-1# config t`	Enters configuration mode.
Step 2	**privilege** *privilege-name* **create** **Example:** `umg-1(config)# privilege security-privilege create`	Creates a new privilege. • *privilege-name*—Label used to identify and configure a new or existing privilege.

	Command or Action	Purpose
Step 3	**privilege** *privilege-name* [**description** *string*] **Example:** `umg-1(config)# privilege security-privilege description administer of system security`	(Optional) Assigns a description to the privilege. • *string*—Description to add to the privilege.
Step 4	**privilege** *privilege-name* **operation** *operation-name* **Example:** `umg-1(config)# privilege security-privilege operation security.configuration`	(Optional) Assigns an operation to the privilege: • *operation-name*—Operation to associate with the privilege.
Step 5	**privilege** *privilege-name* **member** *privilege-name2* **Example:** `umg-1(config)# privilege security-privilege include manage-users`	(Optional) Includes or nests another privilege into this privilege: • *privilege-name2*—Privilege to include or nest into this privilege.
Step 6	**end** **Example:** `umg-1(config)# end`	Exits to privileged EXEC mode.
Step 7	**show operations** **Example:** `umg-1# show operations`	(Optional) Displays information about all operations.
Step 8	**show operation detail** *operation-name* **Example:** `umg-1# show operation detail security.configuration`	(Optional) Displays information about the specified operation: • *operation-name*—Label used to identify and configure a new or existing operation.
Step 9	**show privileges** **Example:** `umg-1# show privilege`	(Optional) Displays information about all privileges.
Step 10	**show privilege detail** *privilege-name* **Example:** `umg-1# show privilege detail sales_vp`	(Optional) Displays information about the specified privilege: • *privilege-name*—Label used to identify and configure a new or existing privilege.

Examples

The following is sample output from the **show operations** command:

```
umg-1# show operations
show operations
group.configuration
security.aaa
security.access
security.password
security.pin
services.configuration
services.exec
```

```
services.manage
software.install
srsx
system.backup
system.configuration
system.debug
system.view
user.configuration
user.password
user.pin

17 total operation(s)
```

The following is sample output from the **show operation detail** command:

```
umg-1# show operation detail user.password
Operation:          user.password
Description:        Set and reset passwords for other users
CLI:
                    config-user-password
                    exec-configure-terminal
                    exec-copy-running-config-startup-config
                    exec-show-user-auth
                    exec-user-password
                    exec-write

6 total command(s)
```

The following is sample output from the **show privileges** command:

```
umg-1# show privileges
ViewRealTimeReports
broadcast
local-broadcast
manage-passwords
manage-users
superuser

6 total privilege(s)
```

Configuring Accounting Event Logging

AAA accounting logs contain information that enables you to easily:

- Audit configuration changes
- Maintain security
- Accurately allocate resources
- Determine who should be billed for the use of resources

You can configure AAA accounting to log the following types of events:

- Logins—All forms of system access, including access to the CLI and GUI when a login is required.
- Logouts—All forms of system access, including access to the CLI and GUI when a login is required before logout.
- Failed logins—Failed login attempts for all forms of system access, including access to the CLI and GUI when a login is required.
- Configuration mode commands—Any changes made to the Cisco UMG configuration using the CLI or GUI.
- EXEC mode commands—Any commands entered in Cisco UMG EXEC mode using the CLI or GUI.
- System startups—System startups, which include information about the system's software version, installed licenses, installed packages, and so on.
- System Shutdowns—System shutdowns, which include information about the system's software version, installed licenses, installed packages, and so on.

In addition to information specific to the type of action performed, the accounting logs also indicate:

- User that authored the action
- Time when the action was executed
- Time when the accounting record was sent to the server

The detailed content of the log entries is explained in the "Examples" section on page 126.

Note Account logging is not performed during the system power-up playback of the startup configuration. When the system boots up, the startup-config commands are not recorded.

Configuring Accounting Event Logging

SUMMARY STEPS

1. **config t**
2. **aaa accounting enable**
3. **aaa accounting event**
4. **login**
5. **logout**
6. **login-fail**
7. **config-commands**
8. **exec-commands**
9. **system-startup**
10. **system-shutdown**
11. **end**
12. **show aaa accounting event**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**config t** **Example:** `umg-1# config t`	Enters configuration mode.
Step 2	**aaa accounting enable** **Example:** `umg-1(config)# aaa accounting enable`	
Step 3	**aaa accounting event** **Example:** `umg-1(config)# aaa accounting event`	
Step 4	**login** **Example:** `umg-1(config)# login`	Enables the logging of logins.
Step 5	**logout** **Example:** `umg-1(config)# logout`	Enables the logging of logouts
Step 6	**login-fail** **Example:** `umg-1(config)# login-fail`	Enables the logging of failed logins.
Step 7	**config-commands** **Example:** `umg-1(config)# config-commands`	Enables the logging of configuration mode commands.
Step 8	**exec-commands** **Example:** `umg-1(config)# exec-commands`	Enables the logging of configuration mode commands.
Step 9	**system-startup** **Example:** `umg-1(config)# system-startup`	Enables the logging of system startups.
Step 10	**system-shutdown** **Example:** `umg-1(config)# system-shutdown`	Enables the logging of system shutdowns.

	Command or Action	Purpose
Step 11	**end** **Example:** `umg-1(config)# end`	Exits to privileged EXEC mode.
Step 12	**show aaa accounting event** `umg-1# show aaa accounting`	(Optional) Displays the AAA accounting events that are designated to be logged.

Examples

The following is sample output from the **show aaa accounting event** command:

```
umg-1# show aaa accounting event
Event               State      Description
login               Enabled    Log accounting events for successful login
logout              Enabled    Log accounting events for user logout
login-fail          Enabled    Log accounting events for failed login attempts
config-commands     Enabled    Log accounting events for any chanes to configuration
exec-commands       Enabled    Log accounting events for execution of commands
system-startup       Enabled    Log accounting events for system startup
system-shutdown     Enabled    Log accounting events for system shutdown
imap                Enabled   Log accounting events for all imap events
```

Configuring Console Authentication

By default, console authentication is disabled, allowing any user logging into the system through the console to have superuser privileges and to log in without providing a username or password.

Therefore, to protect your console from unauthorized access, you must enter the **login** command in config-line mode, as described below.

Note To see whether authentication is enabled for the console, you must view the running configuration.

Specifying Whether the Console Connection is Subject to Authentication

SUMMARY STEPS

1. **config t**
2. **line console**
3. **login**
4. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**config t** **Example:** `umg-1# config t`	Enters configuration mode.
Step 2	**line console** **Example:** `umg-1(config)# line console`	Enters config-line mode to enable you to specify whether the console connection is subject to authentication.
Step 3	**login** **Example:** `umg-1(config-line)# line console`	Requires that any user logging in through the console connection is subject to authentication. The **no** or **default** form of this command disables authentication for the console.
Step 4	**end** **Example:** `umg-1(config)# end`	Exits to privileged EXEC mode.