

Forced Authorization Code

- Information About Forced Authorization Code, on page 1
- Configure Forced Authorization Code, on page 6
- Configuration Example for Forced Authorization Code, on page 10
- Feature Information for Forced Authorization Code, on page 11

Information About Forced Authorization Code

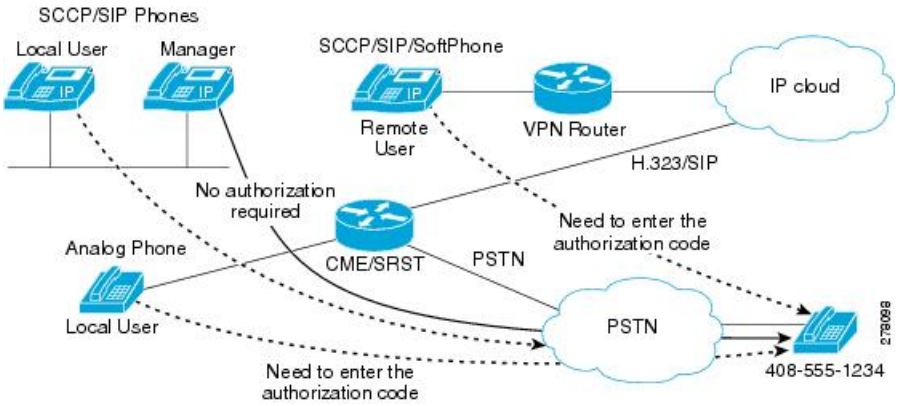
Forced Authorization Code Overview

Cisco Unified CME 8.5 allows you to manage call access and call accounting through the Forced Authorization Code (FAC) feature. The FAC feature regulates the type of call a certain caller may place and forces the caller to enter a valid authorization code on the phone before the call is placed. FAC allows you to track callers dialing non-toll-free numbers, long distance numbers, and also for accounting and billing purposes.

In Cisco Unified CME and Cisco Voice Gateways, devices and endpoints are logically partitioned into different logical partitioning class of restriction (LPCOR) groups. For example, IP phones, Analog phones, PSTN trunks, and IP (h323/SIP) trunks as shown in Figure 1: Forced Authorization Code Network Overview, on page 2, are partitioned into five LPCOR groups under the voice lpcor custom mode, such as:

- voice lpcor custom
- group 10 Manager
- group 11 LocalUser
- group 12 RemoteUser
- group 13 PSTNTrunk
- group 14 IPTrunk

Figure 1: Forced Authorization Code Network Overview



For each group, the LPCOR group policy of a routing endpoint is enhanced to define incoming calls from individual LPCOR groups that are restricted by FAC. A LPCOR group call to a destination is accepted only when a valid FAC is entered. FAC service for a routing endpoint is enabled through the service fac defined in a LPCOR group policy. For more information, see Enable Forced Authorization Code (FAC) on LPCOR Groups, on page 6.

The following are the group policy rules applicable to the PSTNTrunk LPCOR group:

- FAC is required by PSTNTrunk if a call is initiated from either LocalUser or RemoteUser group.
- Any calls from Manager group are allowed to terminate to PSTNTrunk without restriction.
- Any incoming calls from either IPTrunk or PSTNTrunk group are rejected and terminated to PSTNTrunk group.

For information on configuring LPCOR groups and associating LPCOR group with different device types, see Call Restriction Regulations.

FAC Call Flow

FAC is required for an incoming call based on the LPCOR policy defined for the call destination. Once the authentication is finished, the success or failure status and the collected FAC digits are saved to the call detail records (CDRs).

Calls are handled by a new built-in application authorization package which first plays a user-prompt for the caller to enter a username (in digits), then the application plays a passwd-prompt for the caller to collect the password (in digits). The collected username and password digits are then used for FAC, see Define Parameters for Authorization Package, on page 8.

When FAC authentication is successful, the outgoing call setup is continued to the same destination. If FAC authentication fails, the call is then forwarded to the next destination. FAC operations are invoked to the call if FAC service is enabled in the next destination and no valid FAC status is saved for the call.

Any calls failing because of FAC blocking are disconnected with a LPCOR Q.850 disconnect cause code. Once the FAC is invoked for a call, the collected authorization digits and the authentication status information is collected by call active or call history records. You can retrieve the FAC information through the **show call active voice** and **show call history voice** commands.

Forced Authorization Code Specification

The authorization code used for call authentication must follow these specifications:

- The authorization code must be in numeric (0 – 9) format.
- A digit collection operation must be completed if either one of the following conditions occur:
  - maximum number of digits are collected
  - digit input times out
  - a terminating digit is entered

Once digit collection is completed, the authentication is done by either the external Radius server or Cisco Unified CME or Cisco Voice Gateways by using AAA Login Authentication setup. For more information on AAA login authentication methods, see Configuring Authentication.

When authentication is done by local Cisco Unified CME or Cisco Voice Gateways, the **username** *ac-code* **password 0** *password* command is required to authenticate the collected authorization code digits.

FAC data is stored through the CDR and new **AAA fac-digits** and **fac-status** attributes and are supported in a CDR STOP record. This CDR STOP record is formatted for file accounting, RADIUS or Syslog accounting purpose.

FAC Requirement for Different Types of Calls

Table 1: FAC Support for Different Types of Calls, on page 3 shows FAC support for different types of calls.

Table 1: FAC Support for Different Types of Calls

Types of Calls	FAC Behavior for Different Calls
Basic Call	A calls B. B requires A to enter a FAC. A is routed to B only when A enters a valid FAC.
Call Forward All Call Forward Busy	When A (with no FAC) calls B, A is call forwarded to C: • No FAC is required when B enables Call Forward All or Call Forward Busy to C. • FAC is required on A when A is call forwarded to C.
Call Forward No Answer	When A (with no FAC) calls B and A (with FAC) calls C: A calls B: • No FAC is required when A calls B. A is Call Forward No Answer (CFNA) to C. • FAC is required on A when A is call forward to C.

Types of Calls	FAC Behavior for Different Calls
Call Transfer (Blind)	FAC is required, if B calls C and A, and A calls C. **Example:** A calls B. B answers the call. B initiates a blind transfer call to C. A is prompted to enter FAC. A is routed to C only if a valid FAC is entered by A.
Call Transfer (Consultation) Transfer Complete at Alerting State	**1.** FAC is required if B calls C. FAC is not required when A calls C, **Example:** **a.** A calls B. B answers the call and initiates a consultation transfer to C. **b.** B is prompted to enter a FAC and B is not allowed to complete the call transfer when FAC is not completed. **c.** B (the transfer call) is forwarded to C after a valid FAC is entered. B completes the transfer while the transfer call is still ringing on C. A is then transferred to C. **2.** FAC is required if B calls C and A calls C. **Example:** **a.** A calls B. B answers the call and initiates a consultation transfer to C. **b.** B is prompted to enter a FAC and B is not allowed to complete the call transfer when FAC is not completed. **c.** No FAC is required to A, A is then transferred to C. **3.** FAC is not required if B calls C but FAC is required if A calls C. **Example:** **a.** A calls B, B answers the call. **b.** B initiates a consultation transfer to C and completes the transfer. **c.** No FAC required to A, A is then transferred to C.
Transfer Complete at Connected State	**1.** FAC is required when A calls C. **Example:** **a.** A calls B, B answers the call and initiates a consultation transfer to C. **b.** C answers the transfer call and B completes the transfer. **c.** No FAC required to connect to A (including local hairpin calls because the call transfer is complete) and A is connected to C.

Types of Calls	FAC Behavior for Different Calls
Conference Call (Software/Adhoc)	**1.** FAC is not invoked when a call is joined to a conference connection. **2.** FAC is required between A and C, B and C. **Example:** **a.** A calls B, B answers the call and initiates a conference call to C. **b.** B enters a valid authorization code and is routed to C. **c.** C answers the conference call and the conference is complete. **d.** No FAC is required to connect to A and A is joined to a conference connection.
Meetme Conference	**1.** FAC is not invoked for a caller to join the meetme conference. **2.** FAC is required between A and C, B and C. **Example:** **a.** C joins the meetme conference first. **b.** No FAC is required if B joins the same meetme conference. **c.** No FAC is required if C also joins the same meetme conference.
Call Park and Retrieval	**1.** FAC is not invoked for the parked call. **2.** FAC is required if C calls A. **Example:** **a.** A calls B, B answers the call and parks the caller on A. **b.** C retrieves the parked call (A), no FAC is required to reach C, and C is connected to A.
Call Park Restore	**1.** FAC is required if A calls D. **Example:** **a.** A calls B, B answers the call and parks the caller on A. **b.** Parked call (A) is timed out from a call-park slot and is forwarded to D. **c.** No FAC is required for D and the parked call (A) will ring on D.

Types of Calls	FAC Behavior for Different Calls
Group Pickup	1. FAC is not provided if a caller picks up a group call. 2. FAC is required if C calls A. **Example:** a. A calls B, A is ringing on B, and C attempts to pickup call A. b. No FAC is required for C and C is connected to A.
Single Number Redirection (SNR)	FAC is not supported for an SNR call.
Third Party Call Control (3pcc)	FAC is not supported for a three-party call control (3pcc) outgoing call.
Parallel Hunt Groups	FAC is not supported on parallel hunt groups.
Whisper intercom	FAC is not supported for whisper intercom calls.

Configure Forced Authorization Code

Enable Forced Authorization Code (FAC) on LPCOR Groups

Restriction Authenticated FAC data is saved to a call-log from which the authorization code is collected. When a call-forward or blind transfer call scenario triggers a new call due to the SIP notify feature, the same caller is required to enter the authorization code again for FAC authentication.

Warning A FAC pin code must be unique and not the same as an extension number. Cisco Unified CME, Cisco Unified SRST, and Cisco Voice Gateways will not validate whether a collected FAC pin code matches an extension number.

Before you begin

- You must enable the voice lpcor enable command before configuring FAC.
- Trunks (IP and PSTN) must be associated with phones into different LPCOR groups. See Associate a LPCOR Policy with Analog Phone or PSTN Trunk Calls for more information.

SUMMARY STEPS

1. **enable**
2. **configure terminal**

3. **voice lpcor enable**
4. **voice lpcor custom**
5. **group** *number lpcor-group*
6. **exit**
7. **voice lpcor policy** *lpcor-group*
8. **accept** *lpcor-group* **fac**
9. **service** *fac*
10. **end**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**voice lpcor enable** **Example:** `Router(config)# voice lpcor enable`	Enables LPCOR functionality on the Cisco Unified CME router.
Step 4	**voice lpcor custom** **Example:** `Router(config)# voice lpcor custom`	Defines the name and number of LPCOR resource groups on the Cisco Unified CME router.

Step 5	**group** *number lpcor-group*
	Example:
	`Router(cfg-lpcor-custom)#group 10 Manager`
	`Router(cfg-lpcor-custom)#group 11 LocalUser`
	`Router(cfg-lpcor-custom)#group 12 RemoteUser`
	`Router(cfg-lpcor-custom)#group 13 PSTNTrunk`
	`Router(cfg-lpcor-custom)#group 14 IPTrunk`
		Adds a LPCOR resource group to the custom resource list.
		• *number*—Group number of the LPCOR entry. Range: 1 to 64.
		• *lpcor-group*—String that identifies the LPCOR resource group.

Step 6	**exit** **Example:** `Router(conf-voi-serv)# exit`	Exits voice-service configuration mode.
Step 7	**voice lpcor policy** *lpcor-group* **Example:**	Creates a LPCOR policy for a resource group. • *lpcor-group*—Name of the resource group that you defined in Step 5.

	Command or Action	Purpose

	`Router(cfg-lpcor-custom)#group 10 Manager`
	`Router(cfg-lpcor-custom)#group 11 LocalUser`
	`Router(cfg-lpcor-custom)#group 12 RemoteUser`
	`Router(cfg-lpcor-custom)#group 13 PSTNTrunk`
	`Router(cfg-lpcor-custom)#group 14 IPTrunk`	

Step 8	**accept** *lpcor-group* **fac**
	Example:
	`Router(cfg-lpcor-policy)# accept PSTNTrunk fac`
	`Router(cfg-lpcor-policy)# accept Manager fac`
		Allows a LPCOR policy to accept calls associated with the specified resource group.
		• Default: Calls from other groups are rejected; calls from the same resource group are accepted.
		• fac—Valid forced authorization code that the caller needs to enter before the call is routed to its destination.
		• Repeat this command for each resource group whose calls you want this policy to accept.

Step 9	**service** *fac* **Example:** `Router(cfg-lpcor-policy)#service fac`	Enables force authorization code service for a LPCOR group. • Default: No form of the service fac command is the default setting of a LPCOR group policy.
Step 10	**end** **Example:** `Router(config-ephone)# end`	Returns to privileged EXEC mode.

Example

Example:

```
Router# show voice lpcor policy
voice lpcor policy PSTNTrunk (group 13):
service fac is enabled
  ( accept     ) Manager (group 10)
  ( reject     ) LocalUser (group 11)
  ( reject     ) RemoteUser (group 12)
  ( accept     ) PSTNTrunk (group 13)
  ( reject     ) IPTrunk (group 14)
```

Define Parameters for Authorization Package

To define required parameters for user name and password, follow these steps:

SUMMARY STEPS

1. **enable**
2. **configure terminal**

3. **application**
4. **package auth**
5. **param passwd**
6. **param user-prompt** *filename*
7. **param passwd-prompt** *filename*
8. **param max-retries**
9. **param term-digit**
10. **param abort-digit**
11. **param max-digits**
12. **exit**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

Step 3	**application**
	Example:
	`Router(config)#application`
	`Router(config-app)#`
		Enters the application configuration mode.

Step 4	**package auth** **Example:** `Router(config-app)#package auth`	Enters package authorization configuration mode.
Step 5	**param passwd** **Example:** `Router(config-app)#package param passwd 12345`	Character string that defines a predefined password for authorization. **Note** Password digits collection is optional if password digits are predefined in the **param passwd** command.
Step 6	**param user-prompt** *filename* **Example:** `Router(config-app-param)#param user-prompt flash:en_bacd_enter_dest.au`	Allows you to enter the user name parameters required for package authorization for FAC authentication. • **user-prompt** *filename* — Plays an audio prompt requesting the caller to enter a valid username (in digits) for authorization.

	Command or Action	Purpose
Step 7	**param passwd-prompt** *filename* **Example:** `Router(config-app-param)#param passwd-prompt flash:en_welcome.au`	Allows you to enter the password parameters required for package authorization for FAC authentication. • **passwd-prompt** *filename*— Plays an audio prompt requesting the caller to enter a valid password (in digits) for authorization.
Step 8	**param max-retries** **Example:** `Router(config-app-param)#param max-retries 0`	Specifies number of attempts to re-enter an account or a password. • **max-entries**—Value ranges from 0-10, default value is 0.
Step 9	**param term-digit** **Example:** `Router(config-app-param)#param term-digit #`	Specifies digit for terminating an account or a password digit collection.
Step 10	**param abort-digit** **Example:** `Router(config-app-param)#param abort-digit *`	Specifies the digit for aborting username or password digit input. Default value is *.
Step 11	**param max-digits** **Example:** `Router(config-app-param)#param max-digits 32`	Maximum number of digits in a username or password. Range of valid value: 1 - 32. Default value is 32.
Step 12	**exit** **Example:** `Router(conf-app-param)# exit`	Exits package authorization parameter configuration mode.

Configuration Example for Forced Authorization Code

Example for Configuring Forced Authorization Code

This section provides configuration example for Forced Authorization Code.

```
!
gw-accounting aaa
!
aaa new-model
!
aaa authentication login default local
aaa authentication login h323 local
aaa authorization exec h323 local
aaa authorization network h323 local
!
aaa session-id common
!
```

```
voice lpcor enable
voice lpcor custom
group 11 LocalUser
group 12 AnalogPhone
!
voice lpcor policy LocalUser
service fac
accept LocalUser fac
accept AnalogPhone fac
!
voice lpcor policy AnalogPhone
service fac
accept LocalUser fac
accept AnalogPhone fac
!
application
package auth
 param passwd-prompt flash:en_bacd_welcome.au
 param passwd 54321
 param user-prompt flash:en_bacd_enter_dest.au
 param term-digit #
 param abort-digit *
 param max-digits 32
!
username 786 password 0 54321
!
voice-port 0/1/0
station-id name Phone1
station-id number 1235
caller-id enable
!
voice-port 0/1/1
lpcor incoming AnalogPhone
lpcor outgoing AnalogPhone
!
dial-peer voice 11 pots
destination-pattern 99329
port 0/1/1
!
ephone-dn  102  dual-line
number 786786
label HussainFAC
!
!
ephone  102
lpcor type local
lpcor incoming LocalUser
lpcor outgoing LocalUser
device-security-mode none
mac-address 0005.9A3C.7A00
type CIPC
button  1:102
```

Feature Information for Forced Authorization Code

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to https://cfnng.cisco.com/. An account on Cisco.com is not required.

Table 2: Feature Information for Forced Authorization Code

Feature Name	Cisco Unified CME Version	Modification
Forced Authorization Code	8.5	Introduced the FAC feature.