

Basic SIP Configuration

- Overview, on page 1
- SIP Configuration Fundamentals, on page 2
- Configuration Examples, on page 15
- Toll Fraud Prevention, on page 19

Overview

This chapter provides basic configuration information for the following features:

- SIP Register Support
- SIP Redirect Processing
- SIP 300 Multiple Choice Messages
- Interaction with Forking Proxies
- SIP Intra-Gateway Hairpinning

Note H.323 protocol is no longer supported from Cisco IOS XE Bengaluru 17.6.1a onwards. Consider using SIP for multimedia applications.

Finding Support Information for Platforms and Cisco Software Images

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

SIP Register Support

SIP gateways allow registration of E.164 numbers to a SIP proxy or registrar server on behalf of analog phone voice ports (FXS), IP phone virtual voice ports (Ephone-dn Virtual FXS Voice port), and local SCCP phones. By default, SIP gateways do not generate SIP Register messages. The following tasks set up the gateway to register E.164 phone numbers with an external SIP registrar.

Note There are no commands that allow registration with the SIP protocols.

SIP Configuration Fundamentals

Note For help with a procedure, see the verification and troubleshooting sections.

Configure SIP VoIP Services on a CUBE Gateway

Shut Down or Enable VoIP Service on CUBE Gateways

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **voice service voip**
4. **[no] shutdown [forced]**
5. **exit**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enters privileged EXEC mode or any other security level set by a system administrator. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**voice service voip** **Example:** `Router(config)# voice service voip`	Enters voice-service VoIP configuration mode.

	Command or Action	Purpose
Step 4	**[no] shutdown [forced]** **Example:** `Router(config-voi-serv)# shutdown forced`	Shuts down or enables VoIP call services.
Step 5	**exit** **Example:** `Router(config-voi-serv)# exit`	Exits the current mode.

Shut Down or Enable VoIP Submodes on Cisco Gateways

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **voice service voip**
4. **sip**
5. **[no] call service stop** [**forced**] [**maintain-registration**]
6. **exit**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enters privileged EXEC mode or any other security level set by a system administrator. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**voice service voip** **Example:** `Router(config)# voice service voip`	Enters voice-service VoIP configuration mode.
Step 4	**sip** **Example:** `Router(config-voi-serv)# sip`	Enters SIP configuration mode.

	Command or Action	Purpose
Step 5	**[no] call service stop** [**forced**] [**maintain-registration**] **Example:** `Router(conf-serv-sip)# call service stop maintain-registration`	Shuts down or enables VoIP call services for the selected submode.
Step 6	**exit** **Example:** `Router(conf-serv-sip)# exit`	Exits the current mode.

Configure SIP Register Support

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **sip-ua**
4. **registrar** {**dns:** *address* | **ipv4:** *destination-address*} **expires** *seconds* [**tcp**] [**secondary**]
5. **retry register** *number*
6. **timers register** *milliseconds*
7. **exit**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enters privileged EXEC mode or any other security level set by a system administrator. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**sip-ua** **Example:** `Router(config)# sip-ua`	Enters SIP user-agent configuration mode.
Step 4	**registrar** {**dns:** *address* | **ipv4:** *destination-address*} **expires** *seconds* [**tcp**] [**secondary**]	Registers E.164 numbers on behalf of analog phone voice ports (FXS) and IP phone virtual voice ports (EFXS) with

	Command or Action	Purpose
	Example: `Router(config-sip-ua)# registrar ipv4:10.8.17.40 expires 3600 secondary`	an external SIP proxy or SIP registrar server. Keywords and arguments are as follows: • **dns:** *address* --Domain-name server that resolves the name of the dial peer to receive calls. • **ipv4:** *destination-address* --IP address of the dial peer to receive calls. • **expires** *seconds* Default registration time, in seconds. • **tcp** --Sets transport layer protocol to TCP. UDP is the default. • **secondary** --Specifies registration with a secondary SIP proxy or registrar for redundancy purposes. Optional.
Step 5	**retry register** *number* **Example:** `Router(config-sip-ua)# retry register 6`	Use this command to set the total number of SIP Register messages that the gateway should send. The argument is as follows: • *number* --Number of Register message retries. Range: 1–10. Default: 6.
Step 6	**timers register** *milliseconds* **Example:** `Router(config-sip-ua)# timers register 500`	Use this command to set how long the SIP user agent waits before sending register requests. The argument is as follows: • *milliseconds* --Waiting time, in ms. Range: 100–1000. Default: 500.
Step 7	**exit** **Example:** `Router(config-sip-ua)# exit`	Exits the current mode.

Configure SIP Redirect Processing

Configure Call-Redirect Processing

Redirect processing using the **redirection** command is enabled by default. To disable and then reset redirect processing, perform the steps listed in this section:

IP-to-IP call redirection can be enabled globally or on a dial-peer basis. To configure, perform the steps listed in these sections:

Configure Call-Redirect Processing Enhancement

SUMMARY STEPS

1. **enable**

2. **configure terminal**
3. **sip-ua**
4. **no redirection**
5. **redirection**
6. exit

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enters privileged EXEC mode or any other security level set by a system administrator. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**sip-ua** **Example:** `Router(config)# sip-ua`	Enters SIP user-agent configuration mode.
Step 4	**no redirection** **Example:** `Router(config-sip-ua)# no redirection`	Disables redirect handling--causes the gateway to treat incoming 3*xx* responses as 4*xx* error class responses.
Step 5	**redirection** **Example:** `Router(config-sip-ua)# redirection`	Resets call redirection to work as specified in RFC 2543. The command **default redirection**also resets call redirection to work as specified in RFC 2543.
Step 6	exit **Example:** `Router(config-sip-ua)# exit`	Exits the current mode.

Configure Call Redirect to Support Calls on a Specific VoIP Dial Peer

Note

- To specify IP-to-IP call redirection for a specific VoIP dial peer, configure it on an inbound dial peer in dial-peer configuration mode. The default application supports IP-to-IP redirection.
- When IP-to-IP redirection is configured in dial-peer configuration mode, the configuration on the specific inbound dial peer takes precedence over the global configuration that is entered under voice service configuration.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **dial-peer voice** *tag* **voip**
4. **application** *application-name*
5. **redirect ip2ip**
6. **exit**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router# enable`	Enters privileged EXEC mode or any other security level set by a system administrator. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**dial-peer voice** *tag* **voip** **Example:** `Router(config)# dial-peer voice 29 voip`	Use this command to enter dial-peer configuration mode. The argument is as follows: *tag* --Digits that define a particular dial peer. Range: 1to 2,147,483,647 (enter without commas).
Step 4	**application** *application-name* **Example:** `Router(config-dial-peer)# application session`	Enables a specific application on a dial peer. The argument is as follows: *application-name* --Name of the predefined application you wish to enable on the dial peer. For SIP, the default Tcl application (from the Cisco IOS image) is session and can be applied to both VoIP and POTS dial peers. The application must support IP-to-IP redirection.

	Command or Action	Purpose
Step 5	**redirect ip2ip** **Example:** `Router(conf-dial-peer)# redirect ip2ip`	Redirects SIP phone calls to a SIP phone calls on a specific VoIP dial peer using the Cisco IOS voice gateway.
Step 6	**exit** **Example:** `Router(conf-dial-peer)# exit`	Exits the current mode.

Configure SIP Implementation

Minor underlying or minimally configurable features are described in the following sections:

For additional information on SIP implementation enhancements, see "Achieving SIP RFC Compliance."

Interaction with Forking Proxies

Call forking enables the terminating gateway to handle multiple requests and the originating gateway to handle multiple provisional responses for the same call. Call forking is required for the deployment of the *find me/follow me* type of services.

Support for call forking enables the terminating gateway to handle multiple requests and the originating gateway to handle multiple provisional responses for the same call. Interaction with forking proxies applies to gateways acting as a UAC, and takes place when a user is registered to several different locations. When the UAC sends an INVITE message to a proxy, the proxy forks the request and sends it to multiple user agents. The SIP gateway processes multiple 18X responses by treating them as independent transactions under the same call ID. When the relevant dial peers are configured for QoS, the gateway maintains state and initiates RSVP reservations for each of these independent transactions. When it receives an acknowledgment, such as a 200 OK, the gateway accepts the successful acknowledgment and destroys state for all other transactions.

The forking feature sets up RSVP for each transaction *only* if the dial peers are configured for QoS. If not, the calls proceed as best-effort.

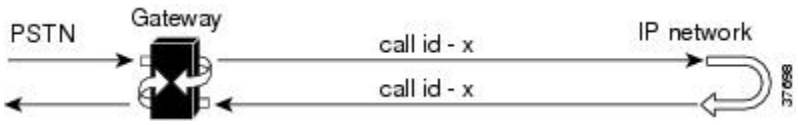
Support for interaction with forking proxies applies only to gateways acting as UACs. It does not apply when the gateway acts as a UAS. In that case, the proxy forks multiple INVITES with the same call ID to the same gateway but with different request URLs.

Also, the forking feature sets up RSVP for each transaction *only* if the dial peers are configured for QoS. If not, the calls proceed as best-effort.

SIP Intra-Gateway Hairpinning

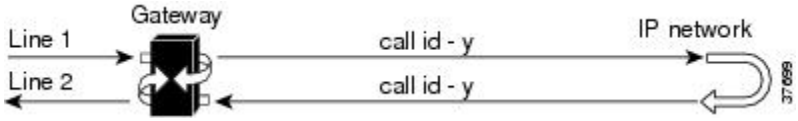
SIP hairpinning is a call routing capability in which an incoming call on a specific gateway is signaled through the IP network and back out the same gateway. This can be a PSTN call routed into the IP network and back out to the PSTN over the same gateway (see the figure below).

Figure 1: PSTN Hairpinning Example



Similarly, SIP hairpinning can be a call signaled from a line (for example, a telephone line) to the IP network and back out to a line on the same access gateway (see the figure below).

Figure 2: Telephone Line Hairpinning Example



With SIP hairpinning, unique gateways for ingress and egress are unnecessary.

SIP supports plain old telephone service (POTS)-to-POTS hairpinning (which means that the call comes in one voice port and is routed out another voice port). It also supports POTS-to-IP call legs and IP-to-POTS call legs. However, it does not support IP-to-IP hairpinning. This means that the SIP gateway cannot take an inbound SIP call and reroute it back to another SIP device using the VoIP dial peers.

Only minimal configuration is required for this feature. To enable hairpinning on the SIP gateway, see the following configuration example for dial peers. Note that:

- The POTS dial peer must have preference 2 defined, and the VoIP dial peer must have preference 1 defined. This ensures that the call is sent out over IP, not Plain Old Telephone Service (POTS).
- The session target is the same gateway because the call is being redirected to it.

```
!
dial-peer voice 53001 pots
 preference 2
 destination-pattern 5300001
 prefix 5300001
!
dial-peer voice 53002 pots
 preference 2
 destination-pattern 5300002
 prefix 5300002
!
dial-peer voice 530011 voip
 preference 1
 destination-pattern 5300001
 session protocol sipv2
 session target ipv4:10.1.1.41
 playout-delay maximum 300
 codec g711alaw
!
dial-peer voice 530022 voip
 preference 1
 destination-pattern 5300002
 session protocol sipv2
 session target ipv4:10.1.1.41
 playout-delay maximum 300
 codec g711alaw
```

Verify CUBE Status

To verify CUBE status and configuration, perform the following steps as appropriate (commands are listed in alphabetical order).

SUMMARY STEPS

1. **show sip service**
2. **show sip-ua register status**
3. **show sip-ua statistics**
4. **show sip-ua status**
5. **show sip-ua timers**

DETAILED STEPS

Procedure

Step 1 **show sip service**

Use this command to display the status of SIP call service on a SIP gateway.

The following sample output shows that SIP call service is enabled:

Example:

```
Router# show sip service
SIP Service is up
```

The following sample output shows that SIP call service was shut down with the **shutdown** command:

Example:

```
Router# show sip service
SIP service is shut globally
under 'voice service voip'
```

The following sample output shows that SIP call service was shut down with the **call service stop** command:

Example:

```
Router# show sip service
SIP service is shut
under 'voice service voip', 'sip' submode
```

The following sample output shows that SIP call service was shut down with the **shutdown forced** command:

Example:

```
Router# show sip service
SIP service is forced shut globally
under 'voice service voip'
```

The following sample output shows that SIP call service was shut down with the **call service stop forced** command:

Example:

```
Router# show sip service
SIP service is forced shut
under 'voice service voip', 'sip' submode
```

Step 2 **show sip-ua register status**

Use this command to display the status of E.164 numbers that a SIP gateway has registered with an external primary SIP registrar.

Example:

```
Router# show sip-ua register status
Line peer expires(sec) registered
4001 20001 596          no
4002 20002 596          no
5100 1     596          no
9998 2     596          no
```

Step 3 **show sip-ua statistics**

Use this command to display response, traffic, and retry SIP statistics, including whether call redirection is disabled.

The following sample shows that four registers were sent:

Example:

```
Router# show sip-ua statistics
SIP Response Statistics (Inbound/Outbound)
    Informational:
      Trying 0/0, Ringing 0/0,
      Forwarded 0/0, Queued 0/0,
      SessionProgress 0/0
     Success:
      OkInvite 0/0, OkBye 0/0,
      OkCancel 0/0, OkOptions 0/0,
      OkPrack 0/0, OkPreconditionMet 0/0,
      OkSubscribe 0/0, OkNOTIFY 0/0,
      OkInfo 0/0, 202Accepted 0/0
      OkRegister 12/49
     Redirection (Inbound only except for MovedTemp(Inbound/Outbound)) :
      MultipleChoice 0, MovedPermanently 0,
      MovedTemporarily 0/0, UseProxy 0,
      AlternateService 0
      Client Error:
      BadRequest 0/0, Unauthorized 0/0,
      PaymentRequired 0/0, Forbidden 0/0,
      NotFound 0/0, MethodNotAllowed 0/0,
      NotAcceptable 0/0, ProxyAuthReqd 0/0,
      ReqTimeout 0/0, Conflict 0/0, Gone 0/0,
      ReqEntityTooLarge 0/0, ReqURITooLarge 0/0,
      UnsupportedMediaType 0/0, BadExtension 0/0,
      TempNotAvailable 0/0, CallLegNonExistent 0/0,
      LoopDetected 0/0, TooManyHops 0/0,
      AddrIncomplete 0/0, Ambiguous 0/0,
      BusyHere 0/0, RequestCancel 0/0,
      NotAcceptableMedia 0/0, BadEvent 0/0,
      SETooSmall 0/0
     Server Error:
      InternalError 0/0, NotImplemented 0/0,
      BadGateway 0/0, ServiceUnavail 0/0,
      GatewayTimeout 0/0, BadSipVer 0/0,
      PreCondFailure 0/0
```

```
     Global Failure:
      BusyEverywhere 0/0, Decline 0/0,
      NotExistAnywhere 0/0, NotAcceptable 0/0
      Miscellaneous counters:
      RedirectRspMappedToClientErr 0
SIP Total Traffic Statistics (Inbound/Outbound)
      Invite 0/0, Ack 0/0, Bye 0/0,
      Cancel 0/0, Options 0/0,
      Prack 0/0, Comet 0/0,
      Subscribe 0/0, NOTIFY 0/0,
      Refer 0/0, Info 0/0
      Register 49/16
Retry Statistics
      Invite 0, Bye 0, Cancel 0, Response 0,
      Prack 0, Comet 0, Reliable1xx 0, NOTIFY 0
      Register 4
SDP application statistics:
Parses: 0, Builds 0
Invalid token order: 0, Invalid param: 0
Not SDP desc: 0, No resource: 0
Last time SIP Statistics were cleared: <never>
```

The following sample output shows the RedirectResponseMappedToClientError status message. An incremented number indicates that 3*xx* responses are to be treated as 4*xx* responses. When call redirection is enabled (default), the RedirectResponseMappedToClientError status message is not incremented.

Example:

```
Router# show sip-ua statistics
SIP Response Statistics (Inbound/Outbound)
    Informational:
      Trying 0/0, Ringing 0/0,
      Forwarded 0/0, Queued 0/0,
      SessionProgress 0/0
    Success:
      OkInvite 0/0, OkBye 0/0,
      OkCancel 0/0, OkOptions 0/0,
      OkPrack 0/0, OkPreconditionMet 0/0,
      OKSubscribe 0/0, OkNotify 0/0,
      202Accepted 0/0
    Redirection (Inbound only):
      MultipleChoice 0, MovedPermanently 0,
      MovedTemporarily 0, UseProxy 0,
      AlternateService 0
    Client Error:
      BadRequest 0/0, Unauthorized 0/0,
      PaymentRequired 0/0, Forbidden 0/0,
      NotFound 0/0, MethodNotAllowed 0/0,
      NotAcceptable 0/0, ProxyAuthReqd 0/0,
      ReqTimeout 0/0, Conflict 0/0, Gone 0/0,
      ReqEntityTooLarge 0/0, ReqURITooLarge 0/0,
      UnsupportedMediaType 0/0, BadExtension 0/0,
      TempNotAvailable 0/0, CallLegNonExistent 0/0,
      LoopDetected 0/0, TooManyHops 0/0,
      AddrIncomplete 0/0, Ambiguous 0/0,
      BusyHere 0/0, RequestCancel 0/0
      NotAcceptableMedia 0/0, BadEvent 0/0
    Server Error:
      InternalError 0/0, NotImplemented 0/0,
      BadGateway 0/0, ServiceUnavail 0/0,
      GatewayTimeout 0/0, BadSipVer 0/0,
      PreCondFailure 0/0
    Global Failure:
      BusyEverywhere 0/0, Decline 0/0,
```

```
      NotExistAnywhere 0/0, NotAcceptable 0/0
    Miscellaneous counters:
      RedirectResponseMappedToClientError 1,
SIP Total Traffic Statistics (Inbound/Outbound)
    Invite 0/0, Ack 0/0, Bye 0/0,
    Cancel 0/0, Options 0/0,
    Prack 0/0, Comet 0/0,
    Subscribe 0/0, Notify 0/0,
    Refer 0/0
Retry Statistics
    Invite 0, Bye 0, Cancel 0, Response 0,
    Prack 0, Comet 0, Reliable1xx 0, Notify 0
SDP application statistics:
 Parses: 0,  Builds 0
 Invalid token order: 0,  Invalid param: 0
 Not SDP desc: 0,  No resource: 0
```

Step 4 **show sip-ua status**

Use this command to display status for the SIP user agent (UA), including whether call redirection is enabled or disabled.

Example:

```
Router# show sip-ua status
SIP User Agent Status
SIP User Agent for UDP : ENABLED
SIP User Agent for TCP : ENABLED
SIP User Agent bind status(signaling): DISABLED
SIP User Agent bind status(media): DISABLED
SIP max-forwards : 6
SIP DNS SRV version: 1 (rfc 2052)
Redirection (3xx) message handling: ENABLED
```

Step 5 **show sip-ua timers**

Use this command to display the current settings for the SIP user-agent (UA) timers.

The following sample output shows the waiting time before a register request is sent--that is, the value that is set with the **timers register** command:

Example:

```
Router# show sip-ua timers
SIP UA Timer Values (millisecs)
trying 500, expires 180000, connect 500, disconnect 500
comet 500, prack 500, rel1xx 500, notify 500
refer 500, register 500
```

Tips to Troubleshoot

For more information on troubleshooting, see the following references:

- "Cisco IOS Voice Troubleshooting and Monitoring Guide"
- Cisco Technical Support at http://www.cisco.com/en/US/support/index.html
- *Cisco IOS Debug Command Reference*
- *Cisco IOS Voice, Video, and Fax Configuration Guide*

- Troubleshooting and Debugging VoIP Call Basics
- VoIP Debug Commands

Note Commands are listed in alphabetical order.

- Verify that SIP-supported codecs are used. Support for codecs varies on different platforms; use the **codec ?** command to determine the codecs available on a specific platform.
- Use the **debug aaa authentication** command to display high-level diagnostics that are related to AAA logins.
- Use the **debug asnl events**command to verify that the SIP subscription server is up. The output displays a pending message if, for example, the client is unsuccessful in communicating with the server.
- Use the **debug ccsip**family of commands for general SIP debugging, including viewing direction-attribute settings and port and network address-translation traces. Use any of the following related commands:
  - **debug ccsip all**--Enables all SIP-related debugging
  - **debug ccsip calls**--Enables tracing of all SIP service-provider interface (SPI) calls
  - **debug ccsip error**--Enables tracing of SIP SPI errors.
  - **debug ccsip events**--Enables tracing of all SIP SPI events
  - **debug ccsip info**--Enables tracing of general SIP SPI information, including verification that call redirection is disabled
  - **debug ccsip media**--Enables tracing of SIP media streams
  - **debug ccsip messages**--Enables all SIP SPI message tracing, such as those that are exchanged between the SIP user-agent client (UAC) and the access server
  - **debug ccsip preauth**--Enables diagnostic reporting of authentication, authorization, and accounting (AAA) preauthentication for SIP calls
  - **debug ccsip states**--Enables tracing of all SIP SPI state tracing
  - **debug ccsip transport**--Enables tracing of the SIP transport handler and the TCP or User Datagram Protocol (UDP) process
- Use the **debug isdn q931**command to display information about call setup and teardown of ISDN network connections (Layer 3) between the local router (user side) and the network.
- Use the **debug kpml** command to enable debug tracing of KeyPad Markup Language (KPML) parser and builder errors.
- Use the **debug radius** command to enable debug tracing of RADIUS attributes.
- Use the **debug rpms-proc preauth** command to enable debug tracing on the Cisco RPMS process for SIP calls.
- Use the debug rtr trace command to trace the execution of an SAA operation.
- Use the **debug voip** family of commands, including the following:

- **debug voip ccapi protoheaders** --Displays messages sent between the originating and terminating gateways. If no headers are being received by the terminating gateway, verify that the **header-passing** command is enabled on the originating gateway.
- **debug voip ivr script**--Displays any errors that might occur when the Tcl script is running.
- **debug voip rtp session named-event 101** --Displays information important to DTMF-relay debugging, if you are using codec types g726r16 or g726r24. Be sure to append the argument *101* to thecommand to prevent the console screen from flooding with messages and all calls from failing.

Sample output for some of these commands follows:

Sample Output for the debug ccsip events Command

- The example shows how the Proxy-Authorization header is broken down into a decoded username and password.

```
Router# debug ccsip events
CCSIP SPI: SIP Call Events tracing is enabled
21:03:21: sippmh_parse_proxy_auth: Challenge is 'Basic'.
21:03:21: sippmh_parse_proxy_auth: Base64 user-pass string is 'MTIzNDU2Nzg5MDEyMzQ1Njou'.
21:03:21: sip_process_proxy_auth: Decoded user-pass string is '1234567890123456:.'.
21:03:21: sip_process_proxy_auth: Username is '1234567890123456'.
21:03:21: sip_process_proxy_auth: Pass is '.'.
21:03:21: sipSPIAddBillingInfoToCcb: sipCallId for billing records =
10872472-173611CC-81E9C73D-F836C2B6@172.18.192.19421:03:21: ****Adding to UAS Request table
```

Sample Output for the debug ccsip info Command

This example shows only the portion of the debug output that shows that call redirection is disabled. When call redirection is enabled (default), there are no debug line changes.

```
Router# debug ccsip info
00:20:32: HandleUdpSocketReads :Msg enqueued for SPI with IPaddr: 172.18.207.10
:5060
00:20:32: CCSIP-SPI-CONTROL:  act_sentinvite_new_message
00:20:32: CCSIP-SPI-CONTROL:  sipSPICheckResponse
00:20:32: sip_stats_status_code
00:20:32: ccsip_get_code_class: !!Call Redirection feature is disabled on the GW
00:20:32: ccsip_map_call_redirect_responses: !!Mapping 302 response to 480
00:20:32:  Roundtrip delay 4 milliseconds for method INVITE
```

Configuration Examples

SIP Register Support Example

```
Current configuration : 3394 bytes
!
version 12.2
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
service internal
```

```
!
memory-size iomem 15
ip subnet-zero
!
no ip domain lookup
!
voice service voip
 redirect ip2ip
sip
 redirect contact order best-match
ip dhcp pool vespa
 network 192.168.0.0 255.255.255.0
 option 150 ip 192.168.0.1
 default-router 192.168.0.1
!
voice call carrier capacity active
!
voice class codec 1
 codec preference 2 g711ulaw
!
no voice hpi capture buffer
no voice hpi capture destination
!
fax interface-type fax-mail
mta receive maximum-recipients 0
!
interface Ethernet0/0
 ip address 10.8.17.22 255.255.0.0
 half-duplex
!
interface FastEthernet0/0
 ip address 192.168.0.1 255.255.255.0
 speed auto
 no cdp enable
 h323-gateway voip interface
 h323-gateway voip id vespa2 ipaddr 10.8.15.4 1718
!
router rip
 network 10.0.0.0
 network 192.168.0.0
!
ip default-gateway 10.8.0.1
ip classless
ip route 0.0.0.0 0.0.0.0 10.8.0.1
no ip http server
ip pim bidir-enable
!
tftp-server flash:SEPDEFAULT.cnf
tftp-server flash:P005B302.bin
call fallback active
!
call application global default.new
call rsvp-sync
!
voice-port 1/0
!
voice-port 1/1
!
mgcp profile default
!
dial-peer voice 1 pots
 destination-pattern 5100
 port 1/0
!
```

```
dial-peer voice 2 pots
 destination-pattern 9998
 port 1/1
!
dial-peer voice 123 voip
 destination-pattern [12]...
 session protocol sipv2
 session target ipv4:10.8.17.42
 dtmf-relay sip-notify
!
gateway
!
sip-ua
 retry invite 3
 retry register 3
 timers register 150
 registrar dns:myhost3.example.com expires 3600
 registrar ipv4:10.8.17.40 expires 3600 secondary
!
telephony-service
 max-dn 10
 max-conferences 4
!
ephone-dn 1
number 4001
!
ephone-dn 2
number 4002
!
line con 0
 exec-timeout 0 0
line aux 0
line vty 0 4
login
line vty 5 15
 login
!
no scheduler allocate
end
```

SIP 300 Multiple Choice Messages Example

This section provides a configuration example showing redirect contact order set to best match.

```
Current configuration : 3394 bytes
!
version 12.2
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
service internal
!
memory-size iomem 15
ip subnet-zero
!
no ip domain lookup
!
voice service voip
 redirect ip2ip
sip
 redirect contact order best-match
ip dhcp pool vespa
```

```
 network 192.168.0.0 255.255.255.0
 option 150 ip 192.168.0.1
 default-router 192.168.0.1
!
voice call carrier capacity active
!
voice class codec 1
 codec preference 2 g711ulaw
!
no voice hpi capture buffer
no voice hpi capture destination
!
fax interface-type fax-mail
mta receive maximum-recipients 0
!
interface Ethernet0/0
 ip address 10.8.17.22 255.255.0.0
 half-duplex
!
interface FastEthernet0/0
 ip address 192.168.0.1 255.255.255.0
 speed auto
 no cdp enable
 h323-gateway voip interface
 h323-gateway voip id vespa2 ipaddr 10.8.15.4 1718
!
router rip
 network 10.0.0.0
 network 192.168.0.0
!
ip default-gateway 10.8.0.1
ip classless
ip route 0.0.0.0 0.0.0.0 10.8.0.1
no ip http server
ip pim bidir-enable
!
tftp-server flash:SEPDEFAULT.cnf
tftp-server flash:P005B302.bin
call fallback active
!
call application global default.new
call rsvp-sync
!
voice-port 1/0
!
voice-port 1/1
!
mgcp profile default
!
dial-peer voice 1 pots
 destination-pattern 5100
 port 1/0
!
dial-peer voice 2 pots
 destination-pattern 9998
 port 1/1
!
dial-peer voice 123 voip
 destination-pattern [12]...
 session protocol sipv2
 session target ipv4:10.8.17.42
 dtmf-relay sip-notify
!
gateway
```

```
!
sip-ua
 retry invite 3
 retry register 3
 timers register 150
 registrar dns:myhost3.example.com expires 3600
 registrar ipv4:10.8.17.40 expires 3600 secondary
!
telephony-service
 max-dn 10
 max-conferences 4
!
ephone-dn 1
number 4001
!
ephone-dn 2
number 4002
!
line con 0
 exec-timeout 0 0
line aux 0
line vty 0 4
login
line vty 5 15
 login
!
no scheduler allocate
end
```

Toll Fraud Prevention

When a Cisco router platform is installed with a voice-capable Cisco IOS Software image, appropriate features must be enabled on the platform to prevent potential toll fraud exploitation by unauthorized users. Deploy these features on all Cisco router Cisco Unified Communications applications that process voice calls, such as Cisco Unified Communications Manager Express (Cisco Unified CME), Cisco Survivable Remote Site Telephony (SRST), Cisco Unified Border Element (CUBE), Cisco IOS-based router and standalone analog and digital PBX and public-switched telephone network (PSTN) gateways, and Cisco contact-center VoiceXML gateways. These features include, but are not limited to, the following:

- Disable secondary dial tone on voice ports--By default, secondary dial tone is presented on voice ports on Cisco router gateways. Use private line automatic ringdown (PLAR) for Foreign Exchange Office (FXO) ports and direct-inward-dial (DID) for T1/E1 ports to prevent secondary dial tone from being presented to inbound callers.
- Cisco router access control lists (ACLs)--Define ACLs to allow only explicitly valid sources of calls to the router or gateway, and therefore to prevent unauthorized SIP calls from unknown parties to be processed and connected by the router or gateway.
- Close unused SIP ports--If either the SIP protocol is not used in your deployment, close the associated protocol ports. If a Cisco voice gateway has dial peers that are configured to route calls outbound to the PSTN using either time division multiplexing (TDM) trunks or IP, close the unused SIP ports so that calls from unauthorized endpoints cannot connect calls. If the protocols are used and the ports must remain open, use ACLs to limit access to legitimate sources.
- Change SIP port 5060--If SIP is actively used, consider changing the port to something other than well-known port 5060.

- SIP registration--If SIP registration is available on SIP trunks, turn on this feature because it provides an extra level of authentication and validation that only legitimate sources can connect calls. If it is not available, ensure that the appropriate ACLs are in place.
- SIP Digest Authentication--If the SIP Digest Authentication feature is available for either registrations or invites, turn on this feature because it provides an extra level of authentication and validation that only legitimate sources can connect calls.
- Explicit incoming and outgoing dial peers--Use explicit dial peers to control the types and parameters of calls that are allowed by the router, especially in IP-to-IP connections on Cisco Unified CME, SRST, and CUBE. Incoming dial peers offer additional control on the sources of calls, and outgoing dial peers on the destinations. Incoming dial peers are always used for calls. If a dial peer is not explicitly defined, the implicit dial peer 0 is used to allow all calls.
- Explicit destination patterns--Use dial peers with more granularity than.T for destination patterns to block disallowed off-net call destinations. Use class of restriction (COR) on dial peers with specific destination patterns to allow even more granular control of calls to different destinations on the PSTN.
- Translation rules--Use translation rules to manipulate dialed digits before calls connect to the PSTN to provide better control over who may dial PSTN destinations. Legitimate users dial an access code and an augmented number for PSTN for certain PSTN (for example, international) locations.
- Tcl and VoiceXML scripts--Attach a Tcl/VoiceXML script to dial peers to do database lookups or additional off-router authorization checks to allow or deny call flows based on origination or destination numbers. Tcl/VoiceXML scripts can also be used to add a prefix to inbound DID calls. If the prefix plus DID matches internal extensions, then the call is completed. Otherwise, a prompt can be played to the caller that an invalid number has been dialed.
- Host name validation--Use the “permit hostname” feature to validate initial SIP Invites that contain a fully qualified domain name (FQDN) host name in the Request Uniform Resource Identifier (Request URI) against a configured list of legitimate source hostnames.
- Dynamic Domain Name Service (DDNS)--If you are using DDNS as the “session target” on dial peers, the actual IP address destination of call connections can vary from one call to the next. Use voice source groups and ACLs to restrict the valid address ranges expected in DDNS responses (which are used later for call setup destinations).

For more configuration guidance, see the “ Cisco IOS Unified Communications Manager Express Toll Fraud Prevention ” paper.