



VoIP Trace

- [Overview, on page 1](#)
- [Prerequisites, on page 3](#)
- [Restrictions, on page 3](#)
- [Benefits of VoIP Trace, on page 3](#)
- [Guide to use VoIP Trace Framework, on page 4](#)
- [Configuration Example for VoIP Trace, on page 5](#)
- [Syslog Messages, on page 8](#)

Overview

VoIP Trace is a Cisco Unified Border Element (CUBE) serviceability framework, which provides a binary trace facility for troubleshooting SIP call issues. The VoIP Trace framework records both successful and failed calls. All call trace data is stored in system memory. In addition to being stored in the system memory, data for calls with IEC errors is also written to the logging buffer.

The VoIP Trace feature is enabled by default and may be used to help troubleshoot issues, even in deployments with high call volumes.

You can configure VoIP Trace for CUBE using the **trace** configuration sub-mode under **voice service voip** configuration mode:

```
router (config)#voice service voip
router(conf-voi-serv)#trace
router(conf-serv-trace)#?
Voip Trace submode commands:
default Set a command to its defaults
exit Exit from voice service voip trace mode
no Negate a command or set its defaults
shutdown Shut Voip Trace debugging
memory-limit Set limit based on memory used
```

Within the VoIP Trace sub-mode (**conf-serv-trace**), you can configure the following CLI commands:

- **memory-limit** [*platform* | *memory*]
- **no [shutdown]**

VoIP Trace is used for event logging and debugging of VoIP parameters. Using the VoIP Trace framework, the following information is recorded:

- SIP messages for SIP trunk to SIP trunk calls

- CCSIP messages, Events, and APIs for CUBE
- SIP Errors
- Call Control (Unified Communication flows processed by CUBE)
- FSM (Finite State Machine) states and events

VoIP Trace monitors and logs SIP signalling and call events in memory as they occur. In the event that a call error is detected, or calls fail with 3xx, 4xx or 5xx cause codes, these event details are written to the logging buffer after the call clears.



Note Traces for error calls are logged at the rate of up to five traces per second.

There's a configurable memory limit allocated for storage of traces in a VoIP Trace framework for CUBE. The maximum memory limit that you can configure for VoIP Trace is 1000 MB. By default, VoIP Trace will use up to 10% of the available platform memory. For example, if CUBE is used on a platform with 8GB of RAM, VoIP Trace will use up to 800MB for trace data. Once the trace memory limit is reached, older traces are overwritten and will no longer be available.

You can configure the trace memory limit using the CLI command **memory-limit** [*platform* | *memory*] under **trace** configuration sub-mode:

```
Router(conf-serv-trace)#memory-limit ?
<10-1000> Specify maximum memory limit in MB
platform Use 10 percent of available memory
```

To display the traces for a call, use the following show command:

- **show voip trace** {*call-id identifier* | *session-id identifier* | *sip-call-id identifier* | *correlator identifier* | *all* | *cover-buffers* | *statistics* | *detail* }

Feature Information

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 www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 1: Feature Information

Feature Name	Releases	Feature Information
Tenant based Filtering for VoIP Trace	Cisco IOS XE Cupertino 17.8.1a	The show voip trace supports tenant based filtering for VoIP Trace.

Feature Name	Releases	Feature Information
VoIP Trace	Cisco IOS XE Amsterdam 17.3.2	VoIP Trace is a CUBE Serviceability framework for Event Logging and Debug Classification. The following commands are introduced: <ul style="list-style-type: none"> • trace • memory-limit [<i>platform</i> <i>memory</i>] • shutdown • show voip trace {<i>call-id identifier</i> <i>session-id identifier</i> <i>sip-call-id identifier</i> <i>correlator identifier</i> all cover-buffers statistics detail }
Cover Buffer Enhancements for VoIP Trace	Cisco IOS XE Dublin 17.12.1a	VoIP trace for SIP messages is enhanced to display cause code in the cover buffer.

Prerequisites

Cisco IOS XE Amsterdam 17.3.2, Cisco IOS XE Bengaluru 17.4.1a or a later release supported by CUBE.

Restrictions

- Unable to trace incoming calls if active calls exhaust the memory-limit.
- It is not possible to trace calls with a single SIP leg using the trace command.

Benefits of VoIP Trace

The following are some of the benefits of VoIP Trace Serviceability framework:

- Enabled by default
- Minimal CLI configuration requirement
- Minimal processing impact
- Automatic call error identification and trace logging based on IEC Errors
- Request-based manual call error identification and trace logging (filtered by call-ID, session-ID and so on)

Guide to use VoIP Trace Framework

The following are some of the usage guidelines for the VoIP trace Serviceability framework.

- Enable or disable your VoIP trace serviceability framework using the following CLI commands:
 - Enable—Configure **trace** under **voice service voip** configuration mode to enable your VoIP trace framework (**trace** is enabled by default).
 - Disable—Configure **shutdown** under **voip trace** configuration mode to disable your VoIP trace framework. To enable VoIP trace after it's disabled, configure the CLI command **no shutdown**.
- If you configure **shutdown** the VoIP trace serviceability framework:
 - Stops tracing for active calls.
 - Deletes all existing traces in the system memory.
- Monitors calls received after enabling VoIP trace.
- Traces stored in memory can be displayed using the show command **show voip trace {call-id identifier | session-id identifier | sip-call-id identifier | correlator identifier | all | cover-buffers | statistics [detail]}**
 - The show command displays traces for both active and disconnected calls.
 - The show command displays information only for the SIP leg.
 - For media forking, VoIP trace also displays information for forked legs.
 - Voice trace can be filtered by tenant tag.
- For the CLI command **memory-limit [platform | memory]**
 - Configure **memory-limit platform** to set 10% of the total memory available to the IOS processor at the time of configuring the command as VoIP trace memory limit.
 - Configure **memory-limit memory** to set a custom VoIP trace memory limit. Range is 10–1000 MB.
 - Configuration of custom memory-limit more than the available platform memory is not allowed. Configuration fails with an error message:

```
router(config)#voice service voip
router(conf-voi-serv)#trace
router(conf-serv-trace)#memory-limit 800
Error: Setting memory-limit more than available platform memory (732 MB) is not
allowed.
```

- Configuration of memory-limit more than the 10% of the available platform memory affects the system performance. Configuration is successful with a warning message:

```
router(config)#voice service voip
router(conf-voi-serv)#trace
router(conf-serv-trace)#memory-limit 100
Warning: Setting memory limit more than 10% of available platform memory (73
MB) will affect system performance.
```

- Reducing the memory-limit from an existing limit **resets** the VoIP trace data. Take copy of the **show voip trace statistics detail** and **show voip trace all** output data before reducing the memory-limit.
 - A confirmation message is displayed when you reduce the memory-limit from an existing limit:


```
Reducing the memory-limit clears all VoIP Trace statistics and data.
If you wish to copy this data first, enter 'no' to cancel,
otherwise enter 'yes' to proceed.
```
- Increasing the memory-limit does not impact the VoIP trace data.

**Note**

- VoIP trace is unable to trace incoming calls if active calls exhaust the memory-limit.
- To change the Timestamps displayed in the VoIP trace, configure the following:

```
router(config)#monitor event-trace timestamps datetime ?
localtime      Use local time zone for timestamps
msec           Include milliseconds in timestamp
show-timezone  Add time zone information to timestamp
```

Configuration Example for VoIP Trace

From Cisco IOS XE Cupertino 17.8.1a, VoIP Trace format has been updated:

- The colon (:) is used as a separator whenever **key: value** pair is displayed.
- The colon (:) is always followed by a space. For example, **Dir: Inbound, Peer-Tag: 10**.

The following is a sample voip trace log snippet from Cisco IOS XE Cupertino 17.8.1a:

```
----- Cover Buffer -----
Search-key      = 808808:8888:1
Timestamp      = *Aug 30 21:21:45.583
Buffer-Id      = 1
CallID         = 1
Peer-CallID    = 2
Correlator     = 1
Called-Number  = 8888
Calling-Number = 808808
SIP CallID     = 1-24132@8.41.17.71
SIP Session ID = aae2281a4212504a96131a56d26bf2ed
GUID          = 1CE915CE8002
-----
0: *Aug 30 21:21:45.583: //1/1CE915CE8002/CUBE_VT/SIP/Msg/ccsipDisplayMsg:
Received: SIP UDP message from 8.41.17.71:5099 to 8.44.25.45:5060
INVITE sip:8888@CUBE.com SIP/2.0
Via: SIP/2.0/UDP 8.41.17.71:5099;branch=z9hG4bK-24132-1-0
From: <sip:808808@8.41.17.71:5099>;tag=1
To: <sip:8888@8.44.25.45:5060>
Call-ID: 1-24132@8.41.17.71
CSeq: 1 INVITE
Contact: sip:808808@8.41.17.71:5099
```

```

Max-Forwards: 70
Subject: Performance Test
Content-Type: application/sdp

v=0
o=user1 53655765 2353687637 IN IP4 8.41.17.71
s=my first call
i=Basic SDP Testing
u=http://www.cisco.com
e=Adithya M <aditm@cisco.com>
p=9880562910
c=IN IP4 8.41.17.71
t=0 0
m=audio 6001 RTP/AVP 8
a=rtpmap:8 PCMA/8000
a=sendrecv

2: *Aug 30 21:21:45.583: //1/1CE915CE8002/CUBE_VT/SIP/FSM/SPI-State-Change: Current State:
  STATE_NONE, Next State: STATE_IDLE, Current Sub-State: STATE_NONE, Next Sub-State: STATE_NONE
3: *Aug 30 21:21:45.588: //1/1CE915CE8002/CUBE_VT/SIP/MISC/Matched Dialpeer: Dir: Inbound,
  Peer-Tag: 251
4: *Aug 30 21:21:45.605: //1/1CE915CE8002/CUBE_VT/SIP/FSM/Offer-Answer: Event:
  E_SIP_INVITE_SDP_RCVD, Current State: S_SIP_EARLY_DIALOG_IDLE, Next State:
  S_SIP_EARLY_DIALOG_OFFER_RCVD
5: *Aug 30 21:21:45.606: //1/1CE915CE8002/CUBE_VT/SIP/FSM/IWF: Event: E_SIP_IWF_EV_RCVD_SDP,
  Current State: S_SIP_IWF_SDP_IDLE, Next State: S_SIP_IWF_SDP_RCVD_AWAIT_PEER_EVENT
6: *Aug 30 21:21:45.607: //1/1CE915CE8002/CUBE_VT/SIP/MISC/Media Stream Parameters: Stream
  Type: voice-only, Stream State: STREAM_ADDING, Negotiated Codec: g711alaw, Negotiated DTMF
  Type: inband-voice, Stream Index: 1

```



Note From Cisco IOS XE Dublin 17.12.1a, VoIP trace format is updated to include cause code in the cover buffer:

The following is a sample VoIP trace for request time out call leg:

```

----- Cover Buffer -----
Search-key      = 808808:6666:4
Timestamp      = Apr 27 09:54:54.491
CallID         = 4
Peer-CallID    = 5
Correlator     = NA
Called-Number  = 6666
Calling-Number = 808808
SIP CallID     = 1-18630@10.1.40.50
SIP SessionID  =
GUID          = 651A3C548005
Tenant        = 0
Cause-code     = recovery on timer expiry (102)
-----
//CUBE receives INVITE message from the client
Received:
INVITE sip:6666@CUBE.com SIP/2.0
Via: SIP/2.0/UDP 10.1.40.50:5082;branch=z9hG4bK-18630-1-0
From: sipp <sip:808808@10.1.40.50:5082>;tag=1
To: sut <sip:6666@10.2.10.10:5060>
Call-ID: 1-18630@10.1.40.50
CSeq: 1 INVITE
Contact: sip:808808@10.1.40.50:5082
Max-Forwards: 70
Subject: Performance Test
Content-Length: 0

```

```
//CUBE reponds with 100 TRYING for the INVITE message received from the client
Sent:
SIP/2.0 100 Trying
Via: SIP/2.0/UDP 10.1.40.50:5082;branch=z9hG4bK-18630-1-0
From: sipp <sip:808808@10.1.40.50:5082>;tag=1
To: sut <sip:6666@10.2.10.10:5060>
Date: Thu, 27 Apr 2023 09:54:54 GMT
Call-ID: 1-18630@10.1.40.50
CSeq: 1 INVITE
Allow-Events: telephone-event
Server: Cisco-SIPGateway/IOS-17.12.20230331.111639
Session-ID: 00000000000000000000000000000000;remote=e521d6e24ce1548b879c4bc674890b01
Content-Length: 0

//CUBE now forwards the INVITE message to the server
Sent:
INVITE sip:6666@10.1.40.50:5083 SIP/2.0
Via: SIP/2.0/UDP 10.2.10.10:5060;branch=z9hG4bK01D3E
From: "sipp " <sip:808808@10.2.10.10>;tag=48289-D57
To: <sip:6666@10.1.40.50>
Date: Thu, 27 Apr 2023 09:54:54 GMT
Call-ID: 6524756F-E41811ED-800B8CD8-819FA3D4@10.2.10.10
Supported: timer,resource-priority,replaces,sdp-anat,X-cisco-srtp-fallback
Min-SE: 1800
Cisco-Guid: 1696218196-3826782701-2147847384-2174723028
User-Agent: Cisco-SIPGateway/IOS-17.12.20230331.111639
Allow: INVITE, OPTIONS, BYE, CANCEL, ACK, PRACK, UPDATE, REFER, SUBSCRIBE, NOTIFY, INFO,
REGISTER
CSeq: 101 INVITE
Timestamp: 1682589294
Contact: <sip:808808@10.2.10.10:5060>
Expires: 180
Allow-Events: telephone-event
Max-Forwards: 69
P-Asserted-Identity: "sipp " <sip:808808@10.2.10.10>
Session-ID: e521d6e24ce1548b879c4bc674890b01;remote=00000000000000000000000000000000
Content-Length: 0

//If CUBE doesn't receive any response from the server after retransmission of INVITE
messages,
then CUBE replies back with '408 Request Timeout' to the client.

Sent:
SIP/2.0 408 Request Timeout
Via: SIP/2.0/UDP 10.1.40.50:5082;branch=z9hG4bK-18630-1-0
From: sipp <sip:808808@10.1.40.50:5082>;tag=1
To: sut <sip:6666@10.2.10.10:5060>;tag=57B32-43B
Date: Thu, 27 Apr 2023 09:54:54 GMT
Call-ID: 1-18630@10.1.40.50
CSeq: 1 INVITE
Allow-Events: telephone-event
Server: Cisco-SIPGateway/IOS-17.12.20230331.111639
Reason: Q.850;cause=102
Session-ID: 00000000000000000000000000000000;remote=e521d6e24ce1548b879c4bc674890b01
Content-Length: 0

//The call is tear down with the ACK message received from the client
Received:
ACK sip:6666@10.2.10.10:5060 SIP/2.0
Via: SIP/2.0/UDP 10.1.40.50:5082;branch=z9hG4bK-18630-1-0
From: sipp <sip:808808@10.1.40.50:5082>;tag=1
To: sut <sip:6666@10.2.10.10:5060>;tag=57B32-43B
Call-ID: 1-18630@10.1.40.50
CSeq: 1 ACK
```

```
Contact: <sip:sipp@10.1.40.50:5082;transport=UDP>  
Max-Forwards: 70  
Subject: Performance Test  
Content-Length: 0
```

Syslog Messages

A syslog message is generated in the following format, when CUBE is configured to register with a SIP trunk:

SIP Trunk Registration Success

```
*Aug 12 08:50:41.922: %SIP-3-REG_INTERNAL: Registration Success.  
Tenant=300;Number=slido123;registrar=dns:slido.com;addr=8.0.0.121;port=6123;connid=4
```

SIP Trunk Registration Failure

```
*Aug 12 08:50:41.876: %SIP-3-REG_INTERNAL: Registration failed.  
Tenant=200;Number=Engineering2541_LGU;registrar=dns:xyz.cisco.com;addr=8.0.0.121;port=9072;connid=3
```



Note

- The syslog message is logged whenever there is change in registration status (from success to failure and vice-versa).
 - If there are consecutive successful or failure responses, only the first successful or failure response is logged.
-