

MediaSense Missing Audio or Call Information



Document ID: 118600

Contributed by Pavan Dave, Cisco TAC Engineer.

Nov 14, 2014

Contents

Introduction

Missing Audio

Example

Missing or Incorrect Call Information

Example 1 – Phone Number Clipped

Example 2 – No Phone Number

Introduction

This document describes two issues you might encounter with Cisco MediaSense and provides troubleshooting information about them. MediaSense is a call recording platform that listens for and records all communications directed at it. If that information, which can be voice or data, is not received or received properly, it might not appear in MediaSense as desired or expected. However, it is often a configuration or network-related issue.

Missing Audio

The first type of call is where the audio is not present but the data is received. In these situations, the problem is typically with a configuration or network-related issue with Access Control Lists (ACLs), Cisco Unified Communications Manager (CUCM), or Cisco Unified Border Element (CUBE).

The best way to verify this type of issue is to make sure that the call control logs are enabled, pull the logs via the Cisco Real-Time Monitoring Tool (RTMT), and have the session ID of the missing audio call log to search.

After you collect the call control logs, open them, search for the session ID, and verify that the size under diskusage is not 0. If the logs show *size="0"*, MediaSense probably did not receive the audio and that is why it is not there.

Example

In this example, the session ID is *78e146437088a93*.

```
0000049583: 10.201.227.136: May 28 2014 11:27:09.022 -0400: %CCBU_COMMON-6-VSMS
HTTP Info: {Thrd=Pool-capture-thread-2800} %[HTTP Response Body=<Session>
  <diskusage>
    <recording name="78e146437088a93-TRACK0" size="0" repository="/
      recordedMedia" />
    <recording name="78e146437088a93-TRACK1" size="0" repository="/
      recordedMedia" />
  </diskusage>
</Session>][HTTP Response Content Type=application/xml][HTTP Response Status
```

Code=200][logId=close-25668]: VSMS Received HTTP Response

When you search, examine the lines that mention diskusage for the particular session ID. In this area, you notice that there is a size in the recording attributes. This example shows that *size="0"*, which means MediaSense did not receive the audio from CUCM or CUBE.

For further troubleshooting tips on missing audio, reference CUCM MediaSense Call Recording Error Troubleshooting.

Missing or Incorrect Call Information

The second type of call is where data is either not present or altered. In these scenarios, the problem is due to configurations on the CUBE or CUCM.

The best way to verify this type of issue is to make sure that the call control logs are enabled and access those via the RTMT. Ensure that you'd to have the session ID of the missing audio call log to search.

Search for a block of text under *CCBU_CALL_CONTROL-6-BORDER_MESSAGE* that contains all of the call information that MediaSense receives, which includes but is not limited to:

- The call's originating location
- The directory numbers (DNs) of the call
- The codec and much more information

If something here does not match what it should be, you might need to analyze the call flow at either CUCM or CUBE in order to determine where the information is altered.

These two examples show these two different issues with missing or incorrect call information.

Example 1 – Phone Number Clipped

In this example, the expected value for session ID *5148fb9543011* shows *19725551234*, but MediaSense only shows *197255512* on Search & Play.

```
0000030499: 10.201.227.36: Oct 10 2014 15:42:16.512 -0400: %CCBU_CALL_CONTROL-6-
BORDER_MESSAGE: {Thrd=Pool-ams-thread-9} %[message_string=HttpPostClient-9:
executing POST http://10.201.227.36:8640/ora/SipAdaptorService/SipAdaptor/
addOrUpdateSession HTTP/1.1
{"sessionData": {
  "callControllerIP": "10.201.227.33",
  "callControllerType": "Cisco-CUCM",
  "endPoints": [
    {
      "clusterid": "StandAloneCluster",
      "conference": false,
      "device": "SEP123456ABCDEF",
      "displayName": "Agent 2102",
      "dn": "2102",
      "startDate": 1412970136508,
      "tracks": [{
        "codec": "PCMU",
        "location": "/recordedMedia",
        "mediaState": "ACTIVE",
        "startDate": 1412970136508,
        "track": 0,
        "type": "AUDIO"
      }],
      "type": "NEAR_END",
```

```

    "xRefci": "37328298"
  },
  {
    "clusterid": "StandAloneCluster",
    "conference": false,
    "device": "S0/SU1/DS1-1@PAVAN-2811",
    "dn": "197255512"",
    "startDate": 1412970136508,
    "tracks": [{
      "codec": "PCMU",
      "location": "/recordedMedia",
      "mediaState": "ACTIVE",
      "startDate": 1412970136508,
      "track": 1,
      "type": "AUDIO"
    }],
    "type": "FAR_END",
    "xRefci": "37328299"
  }
],
"operationType": "ADD",
"recordingServer": "10.201.227.36",
"rtspUrl": "rtsp://10.201.227.36/5148fb9543011",
"sessionName": "5148fb9543011",
"sipServer": "10.201.227.36",
"startDate": 1412970136508,
"state": "ACTIVE",
"version": 7

```

In this situation, notice that MediaSense received the number **19725551234** with the last two digits are stripped off. Since this information comes from CUCM, that is the next place to look in order to determine if CUCM also receives a clipped number from further up the call flow or if this happens on CUCM itself.

Further troubleshooting determined that, in this scenario, CUCM caused the issue, which is described in Cisco bug ID CSCuq20108:

If the From header sent to a recording server exceeds 231 characters, the header will get truncated if escaped characters are found. If the From header contains escaped characters "@", (i.e. %40), the dynamic buffer allocation doesn't account for this resulting in characters getting truncated.

Example 2 – No Phone Number

In this example, the DN is completely absent from a device called **SIP_TRUNK_CVP**.

```

0014107576: 10.201.227.136: Sep 02 2014 16:50:30.484 -0500: %CCBU_CALL_CONTROL-6-
BORDER_MESSAGE: {Thrd=Pool-ams-thread-222081} %[message_string=HttpPostClient-222082:
executing POST http://10.201.227.136:8640/ora/SipAdaptorService/SipAdaptor/
addOrUpdateSession HTTP/1.1
{"sessionData": {
  "callControllerIP": "10.201.227.133",
  "callControllerType": "Cisco-CUCM",
  "endPoints": [
    {
      "conference": false,
      "device": "SEP12356ABCDEF",
      "displayName": "Agent 3102",
      "dn": "3102",
      "startDate": 1409694630483,
      "tracks": [{
        "codec": "PCMU",
        "location": "/recordedMedia",
        "mediaState": "ACTIVE",

```

```

        "startDate": 1409694630483,
        "track": 0,
        "type": "AUDIO"
    }],
    "type": "NEAR_END",
    "xRefci": "65826764"
},
{
    "conference": false,
    "device": "SIP_TRUNK_CVP",
    "dn": "",
    "startDate": 1409694630483,
    "tracks": [{
        "codec": "PCMU",
        "location": "/recordedMedia",
        "mediaState": "ACTIVE",
        "startDate": 1409694630483,
        "track": 1,
        "type": "AUDIO"
    }],
    "type": "FAR_END",
    "xRefci": "65826763"
}
],

```

In this scenario, the logs show that there is no DN information sent to MediaSense, which is very similar to the previous example. In order to further troubleshoot, you should verify that CUCM receives the information correctly, and then check CUBE.