Exploring WebRTC Data and Statistics in WxCC Agent Desktop

Contents

Introduction

This document describes the available WebRTC call data and statistics using built-in browser tools.

Prerequisites

Requirements

Cisco recommends that you have skills and knowledge of Webex Contact Center (WxCC) solution, Chrome, Edge, Firefox browsers and WebRTC project.

Components Used

This document is not restricted to specific software and hardware versions.

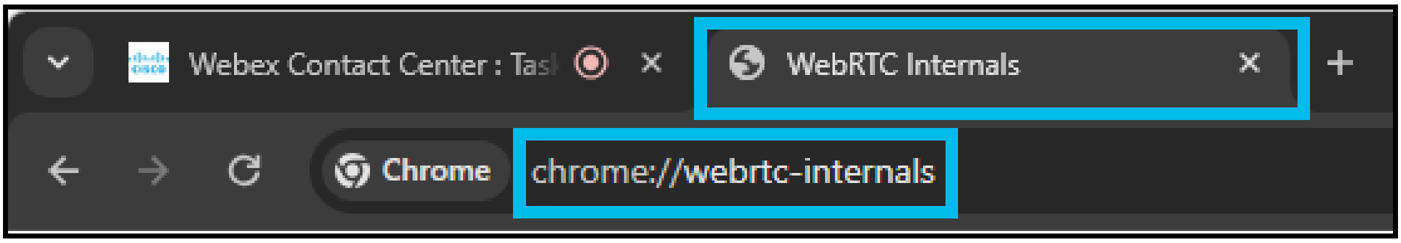
The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. If your network is live, ensure that you understand the potential impact of any command.

Overview

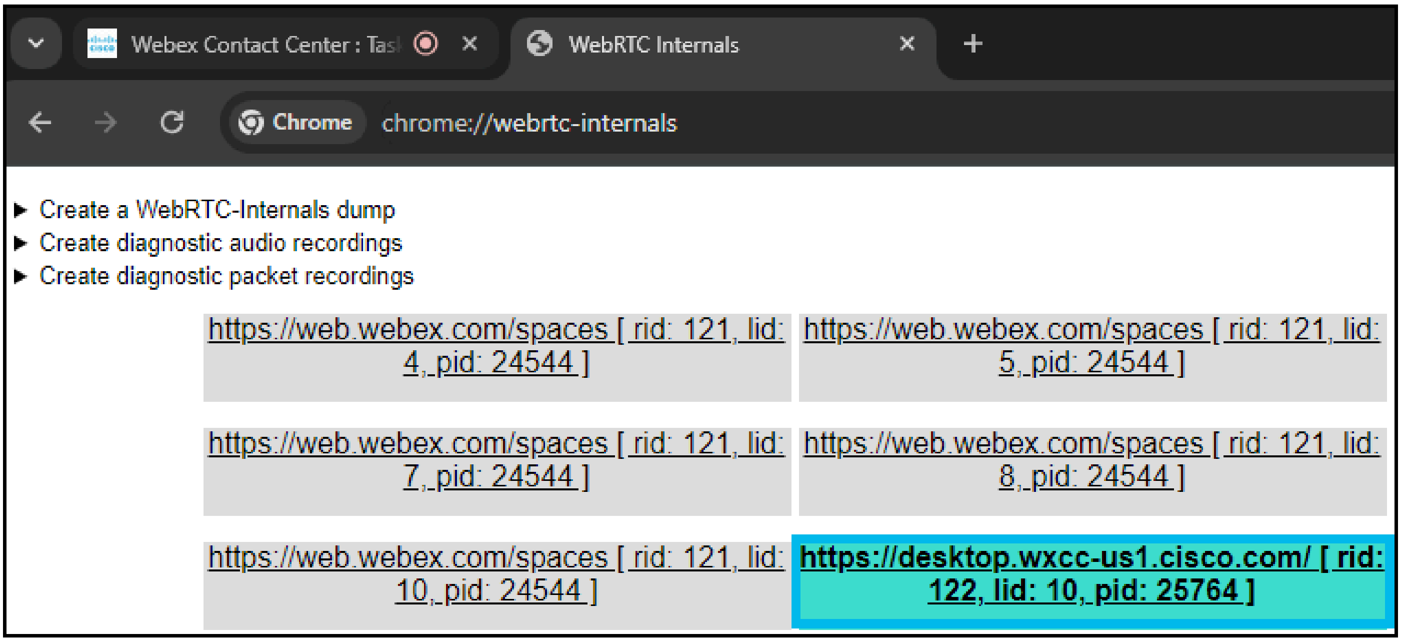
Most web browsers that support WebRTC have built-in WebRTC tools that can be useful for troubleshooting WebRTC related issues.

Chrome

To access WebRTC call data and statistics a new tab in Google Chrome web browser should be opened while on active call and the following URL should be used:
chrome://webrtc-internals.



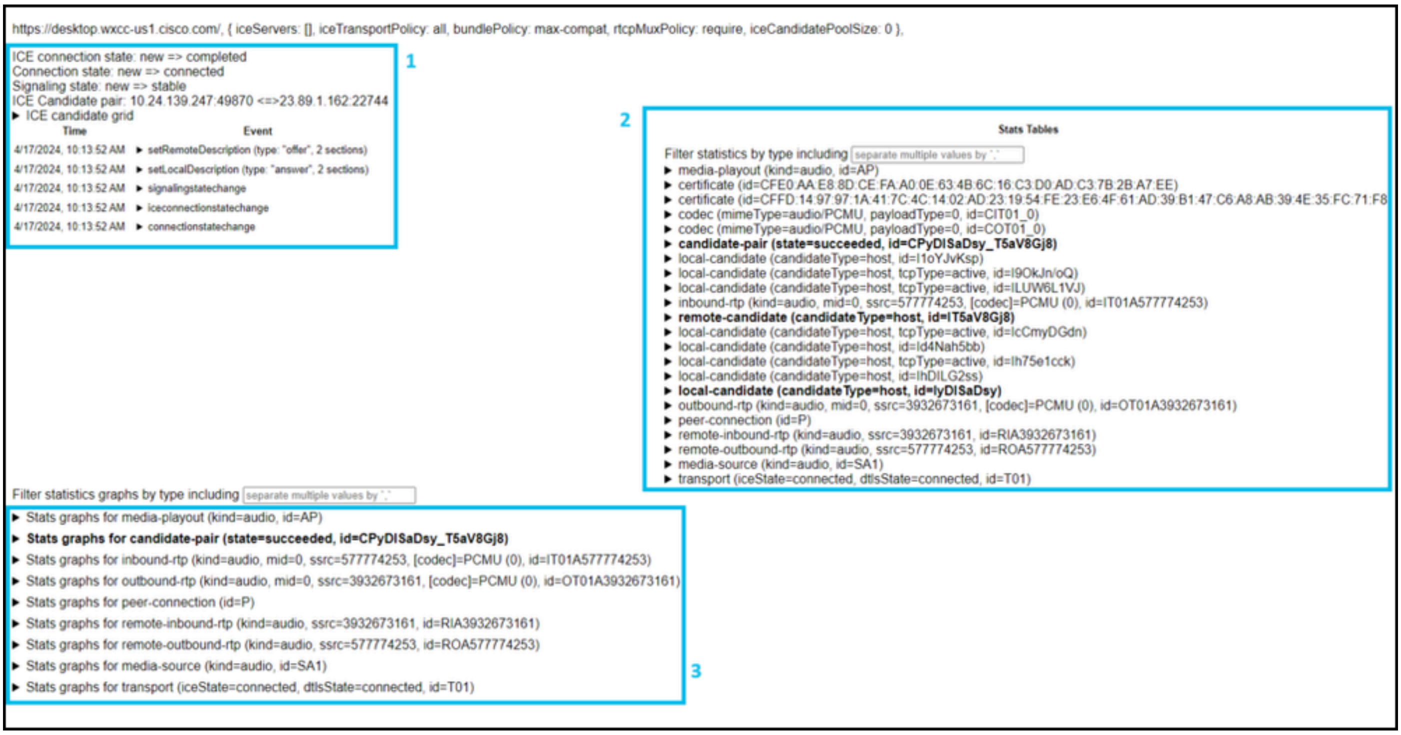
In the WebRTC-internals tab active WebRTC session should be selected (desktop.wxcc-*us1*.cisco.com in its name; URL will depend on the WxCC region).



▶ Create a WebRTC-Internals dump
▶ Create diagnostic audio recordings
▶ Create diagnostic packet recordings

https://web.webex.com/spaces [rid: 121, lid: 4, pid: 24544]
https://web.webex.com/spaces [rid: 121, lid: 5, pid: 24544]
https://web.webex.com/spaces [rid: 121, lid: 7, pid: 24544]
https://web.webex.com/spaces [rid: 121, lid: 8, pid: 24544]
https://web.webex.com/spaces [rid: 121, lid: 10, pid: 24544]
https://desktop.wxcc-us1.cisco.com/ [rid: 122, lid: 10, pid: 25764]

The following information can be seen:
-On the left side **(1)**, a record of the PeerConnection API activities is available. The RTCPeerConnection is the central interface in the WebRTC API. It represents the connection between the local and remote peer and provides all the functions and events necessary to establish the connection (more details can be found at https://webrtc.org/getting-started/overview). These traces **(1)** display interactions with the RTCPeerConnection object and parameters used.
-On the right side **(2)**, the metrics collected via the getStats API about the current WebRTC connection are available.
-On the bottom **(3)**, the graphs produced by utilizing the getStats API **(2)** about the current WebRTC connection are available.



https://desktop.wxcc-us1.cisco.com/, { iceServers: [], iceTransportPolicy: all, bundlePolicy: max-compat, rtcpMuxPolicy: require, iceCandidatePoolSize: 0 },

ICE connection state: new => completed
Connection state: new => connected
Signaling state: new => stable
ICE Candidate pair: 10.24.139.247:49870 <=>23.89.1.162:22744
▶ ICE candidate grid

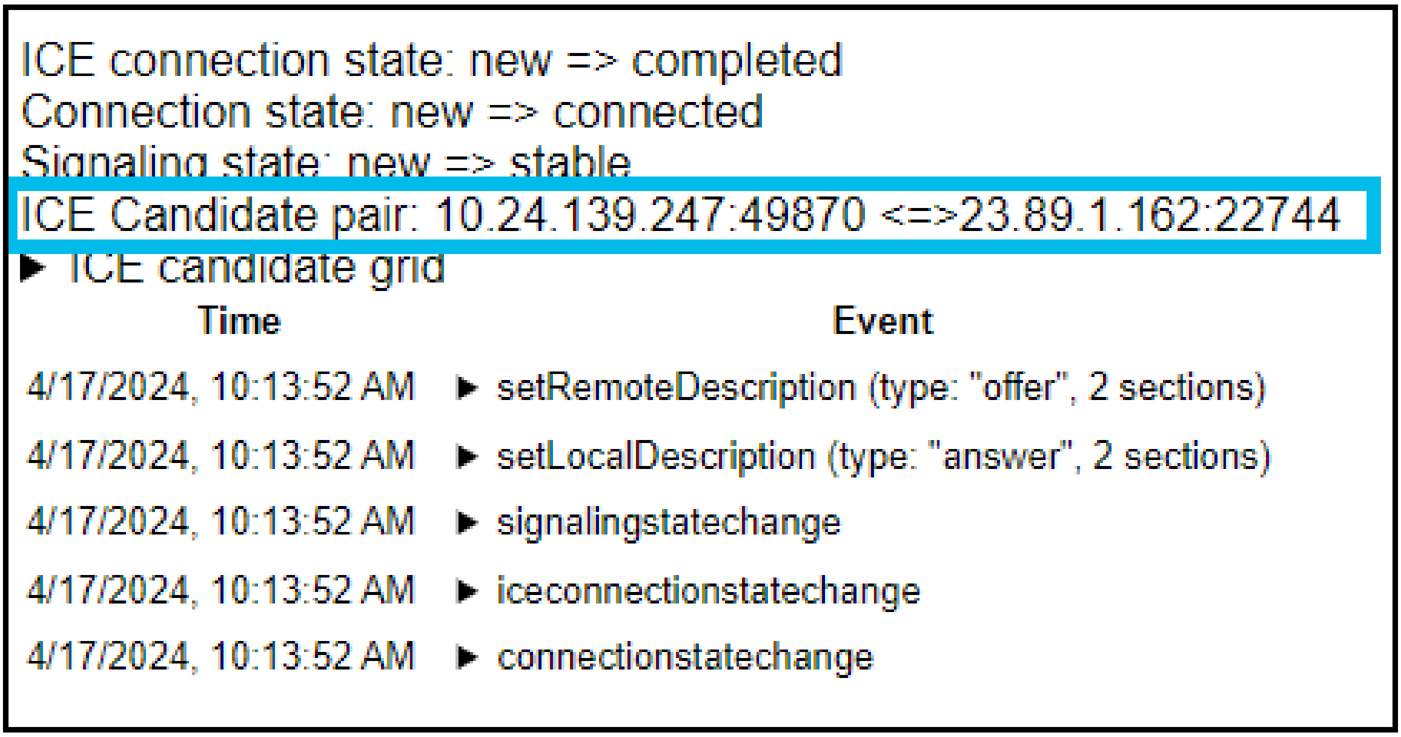
Time	Event
4/17/2024, 10:13:52 AM	▶ setRemoteDescription (type: "offer", 2 sections)
4/17/2024, 10:13:52 AM	▶ setLocalDescription (type: "answer", 2 sections)
4/17/2024, 10:13:52 AM	▶ signalingstatechange
4/17/2024, 10:13:52 AM	▶ iceconnectionstatechange
4/17/2024, 10:13:52 AM	▶ connectionstatechange

Stats Tables

Filter statistics by type including separate multiple values by ','
▶ media-playout (kind=audio, id=AP)
▶ certificate (id=CFE0:AA:E8:8D:CE:FA:A0:0E:63:4B:6C:16:C3:D0:AD:C3:7B:2B:A7:EE)
▶ certificate (id=CFFD:14:97:97:1A:41:7C:4C:14:02:AD:23:19:54:FE:23:E6:4F:61:AD:39:B1:47:C6:A8:AB:39:4E:35:FC:71:F8
▶ codec (mimeType=audio/PCMU, payloadType=0, id=CIT01_0)
▶ codec (mimeType=audio/PCMU, payloadType=0, id=COT01_0)
▶ **candidate-pair (state=succeeded, id=CPyDISaDsy_T5aV8Gj8)**
▶ local-candidate (candidateType=host, id=I1oYJvKsp)
▶ local-candidate (candidateType=host, tcpType=active, id=I9OkJn/oQ)
▶ local-candidate (candidateType=host, tcpType=active, id=ILUW6L1VJ)
▶ inbound-rtp (kind=audio, mid=0, ssrc=577774253, [codec]=PCMU (0), id=IT01A577774253)
▶ **remote-candidate (candidateType=host, id=IT5aV8Gj8)**
▶ local-candidate (candidateType=host, tcpType=active, id=IcCmyDGdn)
▶ local-candidate (candidateType=host, id=Id4Nah5bb)
▶ local-candidate (candidateType=host, tcpType=active, id=Ih75e1cck)
▶ local-candidate (candidateType=host, id=IhDILG2ss)
▶ **local-candidate (candidateType=host, id=IyDISaDsy)**
▶ outbound-rtp (kind=audio, mid=0, ssrc=3932673161, [codec]=PCMU (0), id=OT01A3932673161)
▶ peer-connection (id=P)
▶ remote-inbound-rtp (kind=audio, ssrc=3932673161, id=RIA3932673161)
▶ remote-outbound-rtp (kind=audio, ssrc=577774253, id=ROA577774253)
▶ media-source (kind=audio, id=SA1)
▶ transport (iceState=connected, dtlsState=connected, id=T01)

Filter statistics graphs by type including separate multiple values by ','
▶ Stats graphs for media-playout (kind=audio, id=AP)
▶ **Stats graphs for candidate-pair (state=succeeded, id=CPyDISaDsy_T5aV8Gj8)**
▶ Stats graphs for inbound-rtp (kind=audio, mid=0, ssrc=577774253, [codec]=PCMU (0), id=IT01A577774253)
▶ Stats graphs for outbound-rtp (kind=audio, mid=0, ssrc=3932673161, [codec]=PCMU (0), id=OT01A3932673161)
▶ Stats graphs for peer-connection (id=P)
▶ Stats graphs for remote-inbound-rtp (kind=audio, ssrc=3932673161, id=RIA3932673161)
▶ Stats graphs for remote-outbound-rtp (kind=audio, ssrc=577774253, id=ROA577774253)
▶ Stats graphs for media-source (kind=audio, id=SA1)
▶ Stats graphs for transport (iceState=connected, dtlsState=connected, id=T01)

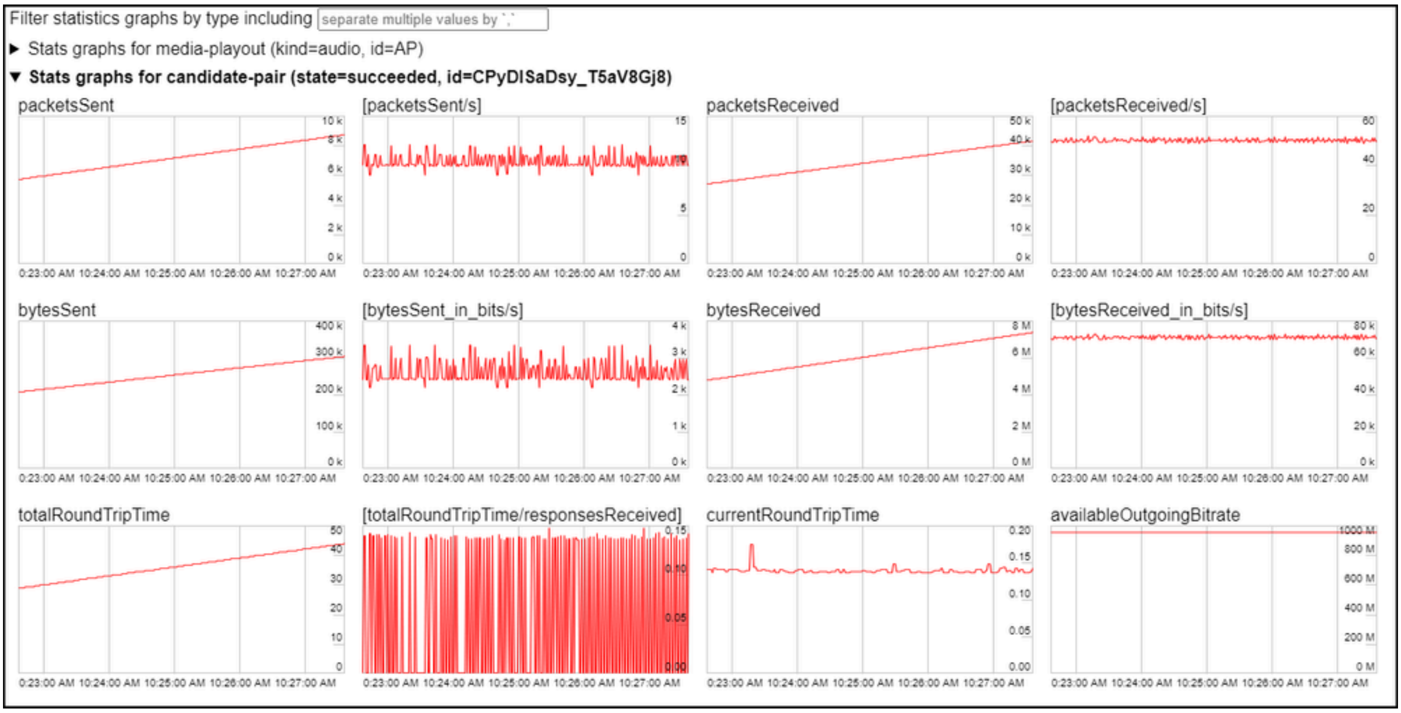
Local and remote IPs and ports in use for this particular WebRTC connection can be seen under traces of the PeerConnection API calls.



ICE connection state: new => completed
Connection state: new => connected
Signaling state: new => stable
ICE Candidate pair: 10.24.139.247:49870 <=>23.89.1.162:22744
▶ ICE candidate grid

Time	Event
4/17/2024, 10:13:52 AM	▶ setRemoteDescription (type: "offer", 2 sections)
4/17/2024, 10:13:52 AM	▶ setLocalDescription (type: "answer", 2 sections)
4/17/2024, 10:13:52 AM	▶ signalingstatechange
4/17/2024, 10:13:52 AM	▶ iceconnectionstatechange
4/17/2024, 10:13:52 AM	▶ connectionstatechange

Various live call statistics (that is. sent packets, received packets, latency, dropped packets and so on.) related graphs can be seen at the bottom (bolded text).

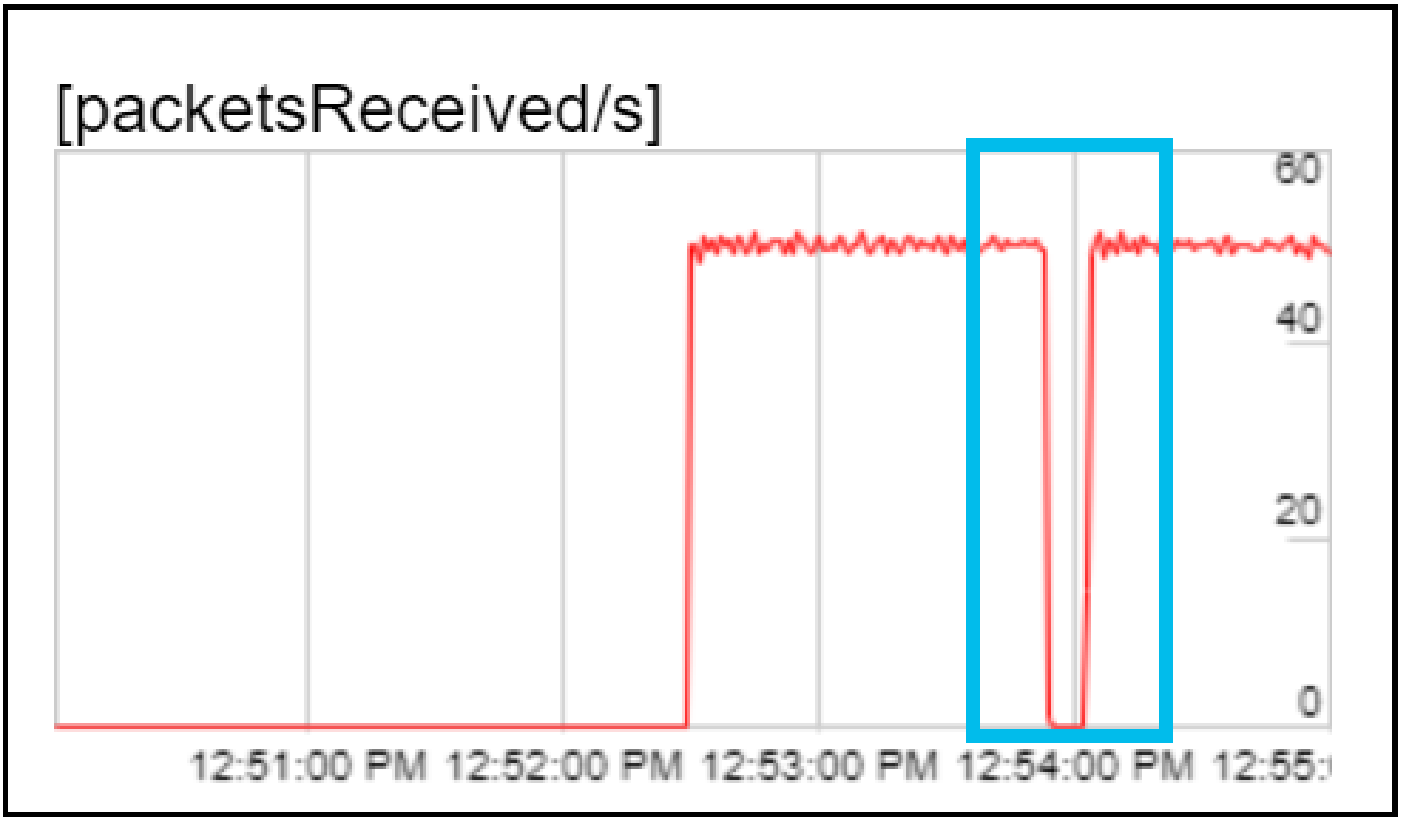


Filter statistics graphs by type including separate multiple values by `,`

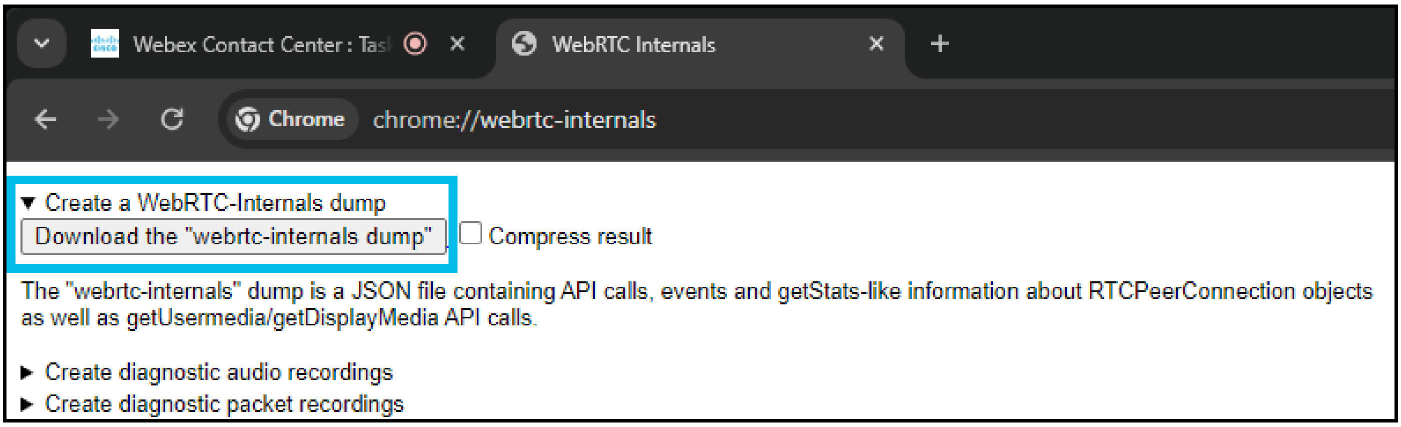
▶ Stats graphs for media-playout (kind=audio, id=AP)

▼ **Stats graphs for candidate-pair (state=succeeded, id=CPyDISaDsy_T5aV8Gj8)**

As an example - missing incoming packets would cause missing audio:



WebRTC log dump can be retrieved by clicking 'Create a WebRTC-Internals dump' and clicking 'Download the "webrtc-internals dump" option.



▼ Create a WebRTC-Internals dump

Download the "webrtc-internals dump" □ Compress result

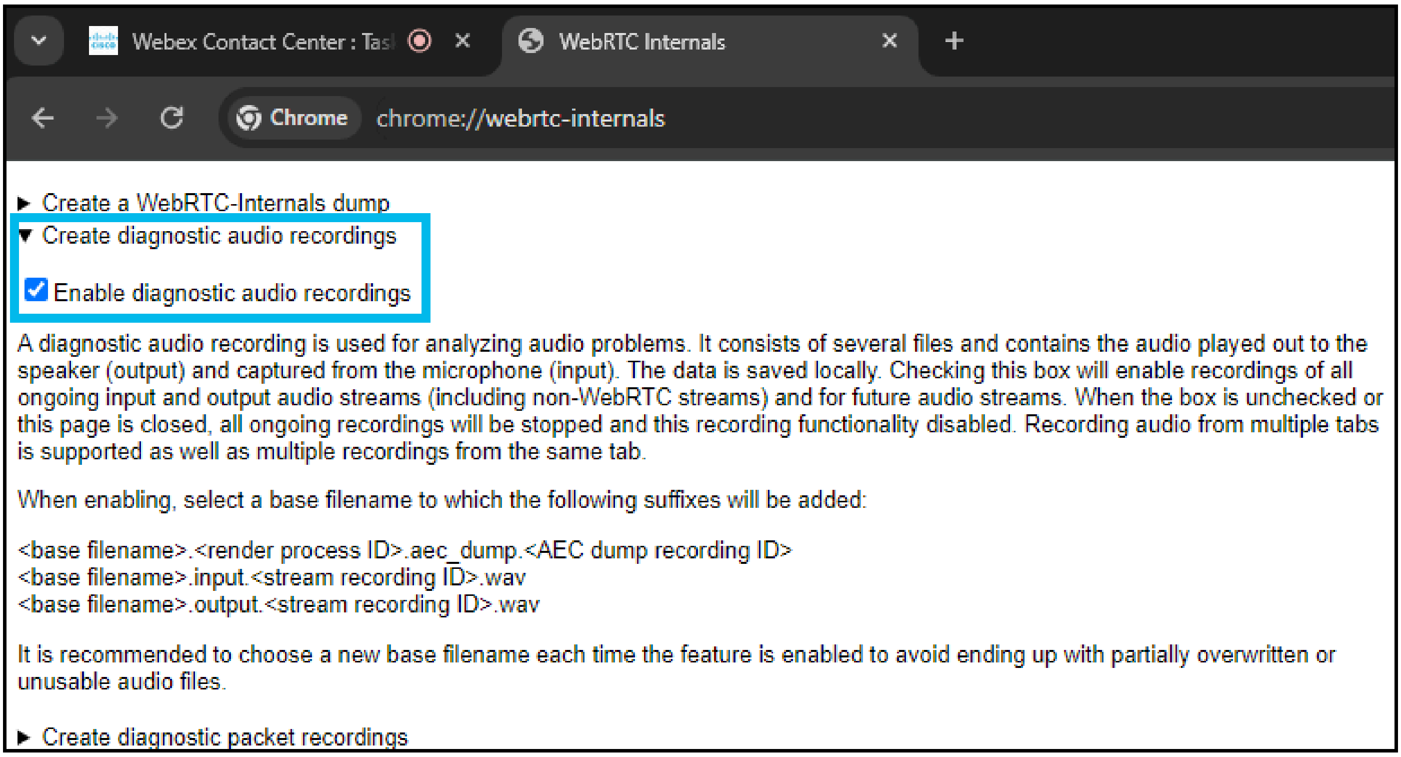
The "webrtc-internals" dump is a JSON file containing API calls, events and getStats-like information about RTCPeerConnection objects as well as getUsermedia/getDisplayMedia API calls.

▶ Create diagnostic audio recordings

▶ Create diagnostic packet recordings

Note: *WebRTC log dump file can be parsed and analyzed using specific tools, that is.* https://fippo.github.io/webrtc-dump-importer/

WebRTC audio can also be captured for diagnostic purposes.



▶ Create a WebRTC-Internals dump

▼ Create diagnostic audio recordings

☑ Enable diagnostic audio recordings

A diagnostic audio recording is used for analyzing audio problems. It consists of several files and contains the audio played out to the speaker (output) and captured from the microphone (input). The data is saved locally. Checking this box will enable recordings of all ongoing input and output audio streams (including non-WebRTC streams) and for future audio streams. When the box is unchecked or this page is closed, all ongoing recordings will be stopped and this recording functionality disabled. Recording audio from multiple tabs is supported as well as multiple recordings from the same tab.

When enabling, select a base filename to which the following suffixes will be added:

<base filename>.<render process ID>.aec_dump.<AEC dump recording ID>
<base filename>.input.<stream recording ID>.wav
<base filename>.output.<stream recording ID>.wav

It is recommended to choose a new base filename each time the feature is enabled to avoid ending up with partially overwritten or unusable audio files.

▶ Create diagnostic packet recordings