

WebRTC-Daten und -Statistiken im WxCC Agent Desktop

Inhalt

[Einleitung](#)

[Voraussetzungen](#)

[Anforderungen](#)

[Verwendete Komponenten](#)

[Überblick](#)

[Chrom](#)

Einleitung

In diesem Dokument werden die verfügbaren WebRTC-Anrufrufen und -statistiken mithilfe integrierter Browser-Tools beschrieben.

Voraussetzungen

Anforderungen

Cisco empfiehlt, dass Sie über Kenntnisse und Fähigkeiten in den Bereichen WebEx Contact Center (WxCC)-Lösung, Chrome, Edge, Firefox-Browser und WebRTC-Projekt verfügen.

Verwendete Komponenten

Dieses Dokument ist nicht auf bestimmte Software- und Hardware-Versionen beschränkt.

Die Informationen in diesem Dokument beziehen sich auf Geräte in einer speziell eingerichteten Testumgebung. Alle Geräte, die in diesem Dokument benutzt wurden, begannen mit einer gelöschten (Nichterfüllungs) Konfiguration. Wenn Ihr Netzwerk in Betrieb ist, stellen Sie sicher, dass Sie die möglichen Auswirkungen aller Befehle kennen.

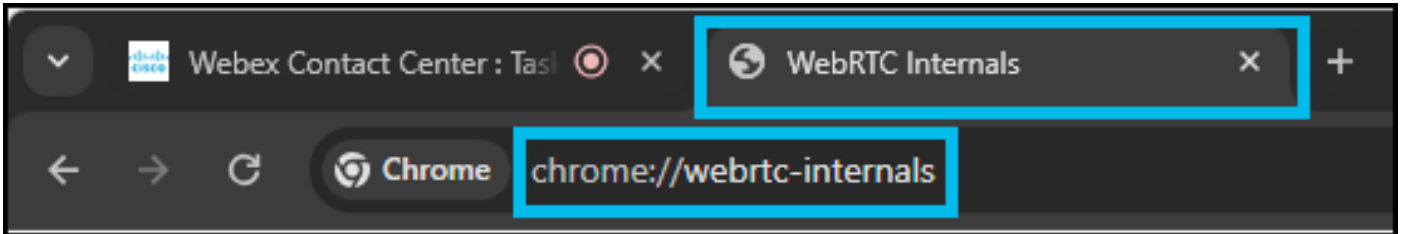
Überblick

Die meisten Webbrowser, die WebRTC unterstützen, verfügen über integrierte WebRTC-Tools, die zur Behebung von WebRTC-Problemen nützlich sein können.

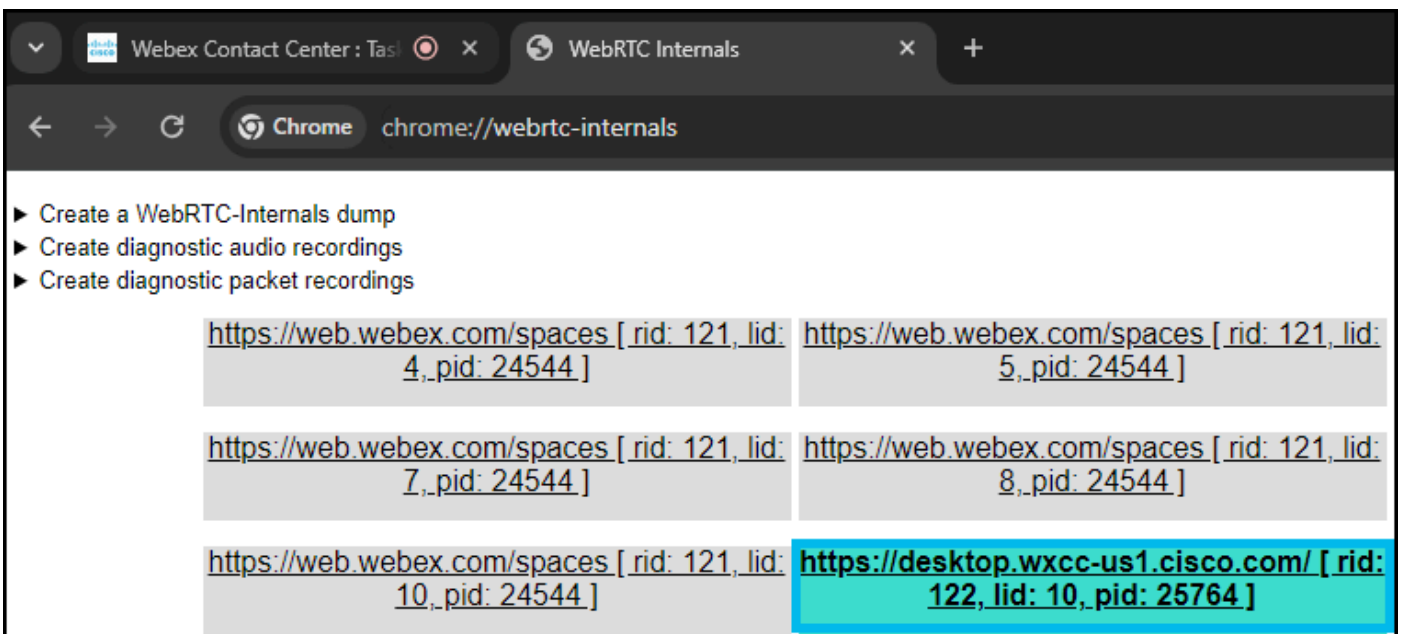
Chrom

Um auf WebRTC-Anrufrdaten und -Statistiken zuzugreifen, sollte im Google Chrome-Webbrowser während des aktiven Anrufs eine neue Registerkarte geöffnet und die folgende URL verwendet werden:

chrome://webrtc-internals



Auf der Registerkarte WebRTC-internals sollte eine aktive WebRTC-Sitzung ausgewählt werden (desktop.wxcc-us1.cisco.com in ihrem Namen; URL hängt von der WxCC-Region ab).



Folgende Informationen sind zu sehen:

- Auf der linken Seite (1) ist ein Datensatz der PeerConnection-API-Aktivitäten verfügbar. Die RTCPeerConnection ist die zentrale Schnittstelle der WebRTC API. Sie stellt die Verbindung zwischen dem lokalen und dem Remote-Peer dar und stellt alle Funktionen und Ereignisse bereit, die zum Herstellen der Verbindung erforderlich sind (weitere Details finden Sie unter <https://webrtc.org/getting-started/overview>). Diese Traces (1) zeigen Interaktionen mit dem RTCPeerConnection-Objekt und den verwendeten Parametern an.
- Auf der rechten Seite (2) sind die über die getStats-API gesammelten Metriken zur aktuellen WebRTC-Verbindung verfügbar.
- Unten (3) sind die Graphen, die mit der getStats-API (2) über die aktuelle WebRTC-Verbindung erzeugt werden, verfügbar.

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

Filter statistics graphs by type including

- ▶ Stats graphs for media-playout (kind=audio, id=AP)
- ▶ **Stats graphs for candidate-pair (state=succeeded, id=CPyDiSaDsy_T5aV8Gj8)**
- ▶ Stats graphs for inbound-rtcp (kind=audio, mid=0, ssrc=577774253, [codec]=PCMU (0), id=IT01A57774253)
- ▶ Stats graphs for outbound-rtcp (kind=audio, mid=0, ssrc=3932673161, [codec]=PCMU (0), id=OT01A3932673161)
- ▶ Stats graphs for peer-connection (id=P)
- ▶ Stats graphs for remote-inbound-rtcp (kind=audio, ssrc=3932673161, id=RIA3932673161)
- ▶ Stats graphs for remote-outbound-rtcp (kind=audio, ssrc=577774253, id=ROA57774253)
- ▶ Stats graphs for media-source (kind=audio, id=SA1)
- ▶ Stats graphs for transport (iceState=connected, dtlsState=connected, id=T01)

Stats Tables

Filter statistics by type including

- ▶ media-playout (kind=audio, id=AP)
- ▶ certificate (id=CFE0AAE88DCEFAA00E634B6C16C3D0AD.C37B2BA7EE)
- ▶ 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=ItoYJvKsp)
- ▶ local-candidate (candidateType=host, tcpType=active, id=I9OkJn/oQ)
- ▶ local-candidate (candidateType=host, tcpType=active, id=ILUW6L1VJ)
- ▶ inbound-rtcp (kind=audio, mid=0, ssrc=577774253, [codec]=PCMU (0), id=IT01A57774253)
- ▶ **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-rtcp (kind=audio, mid=0, ssrc=3932673161, [codec]=PCMU (0), id=OT01A3932673161)
- ▶ peer-connection (id=P)
- ▶ remote-inbound-rtcp (kind=audio, ssrc=3932673161, id=RIA3932673161)
- ▶ remote-outbound-rtcp (kind=audio, ssrc=577774253, id=ROA57774253)
- ▶ media-source (kind=audio, id=SA1)
- ▶ transport (iceState=connected, dtlsState=connected, id=T01)

Lokale und Remote-IPs und -Ports, die für diese spezielle WebRTC-Verbindung verwendet werden, sind unter den Ablaufverfolgungen der PeerConnection-API-Aufrufe zu sehen.

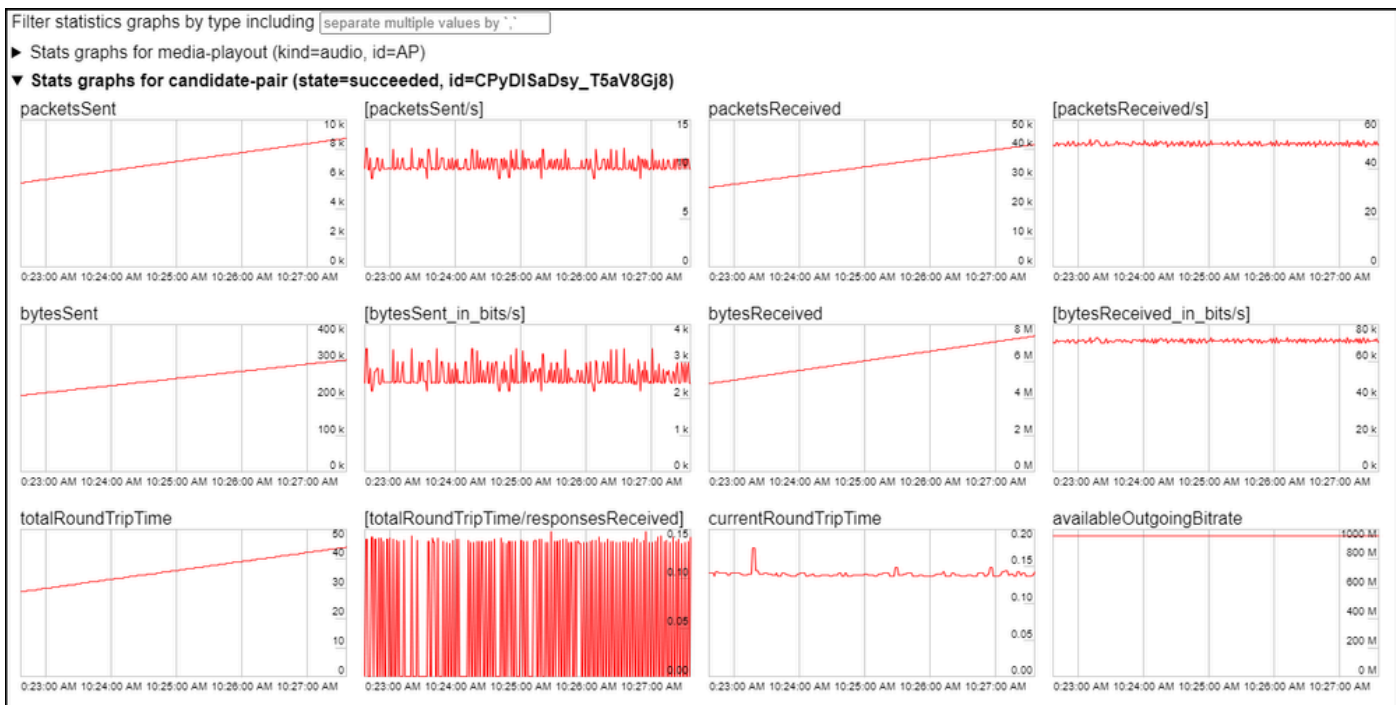
ICE connection state: new => completed
 Connection state: new => connected
 Signaling state: new => stable

ICE Candidate pair: 10.24.139.247:49870 <=>23.89.1.162:22744

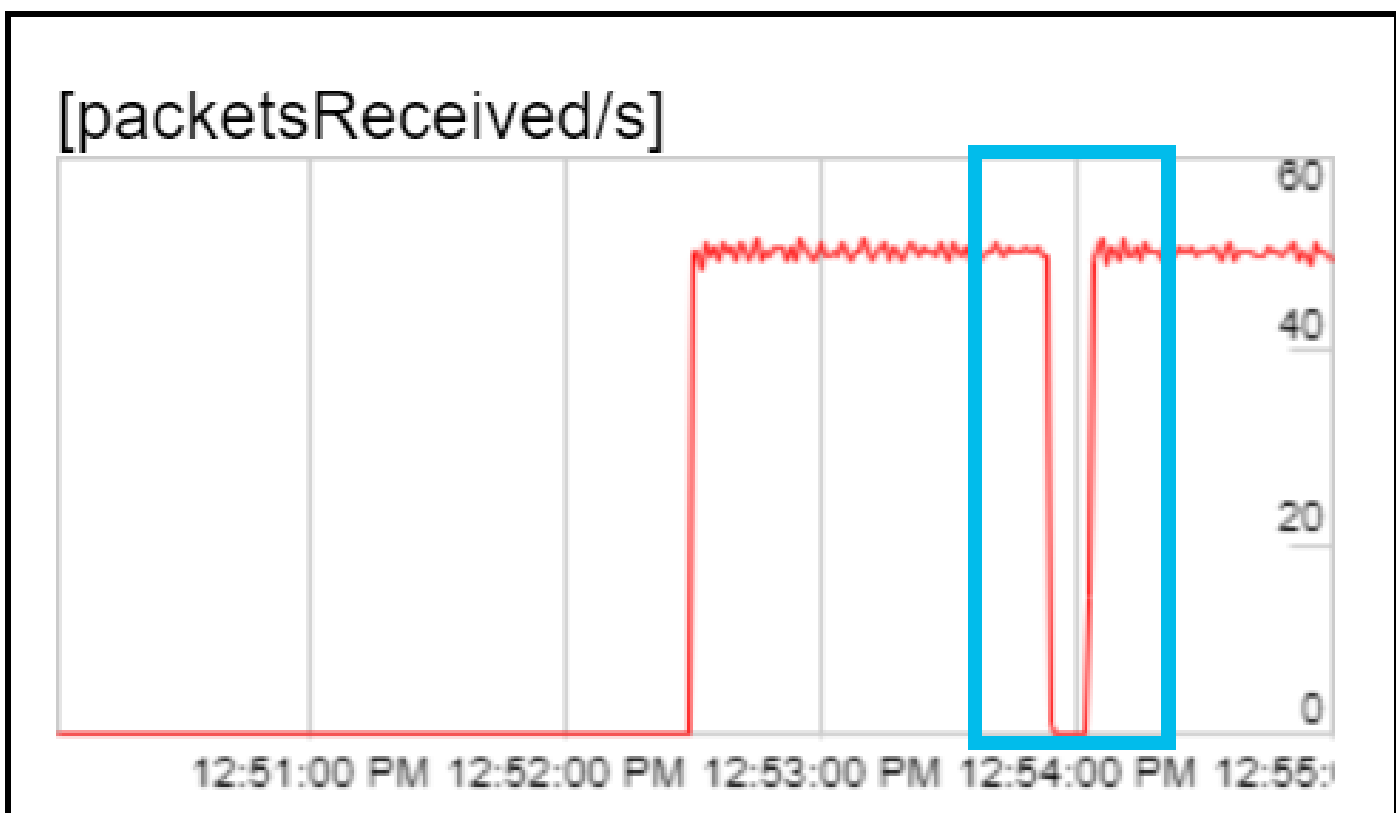
▶ ICE candidate grid

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

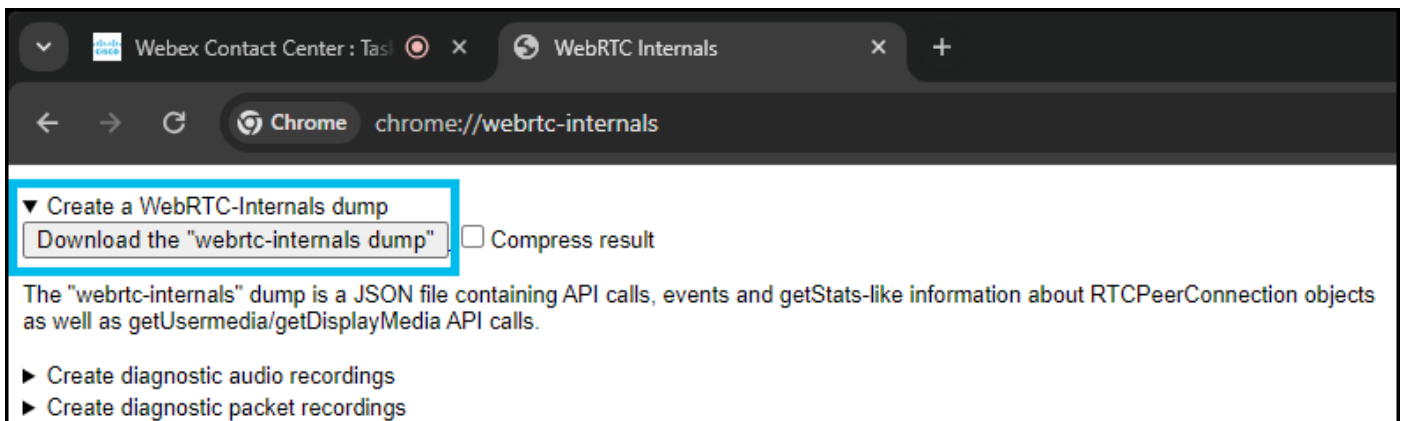
Verschiedene Live-Anrufstatistiken (d. h. gesendete Pakete, empfangene Pakete, Latenz, verworfene Pakete usw.) Diagramme sind unten zu sehen (fett formatierter Text).



Beispiel: Wenn eingehende Pakete fehlen, kann Audio fehlen:

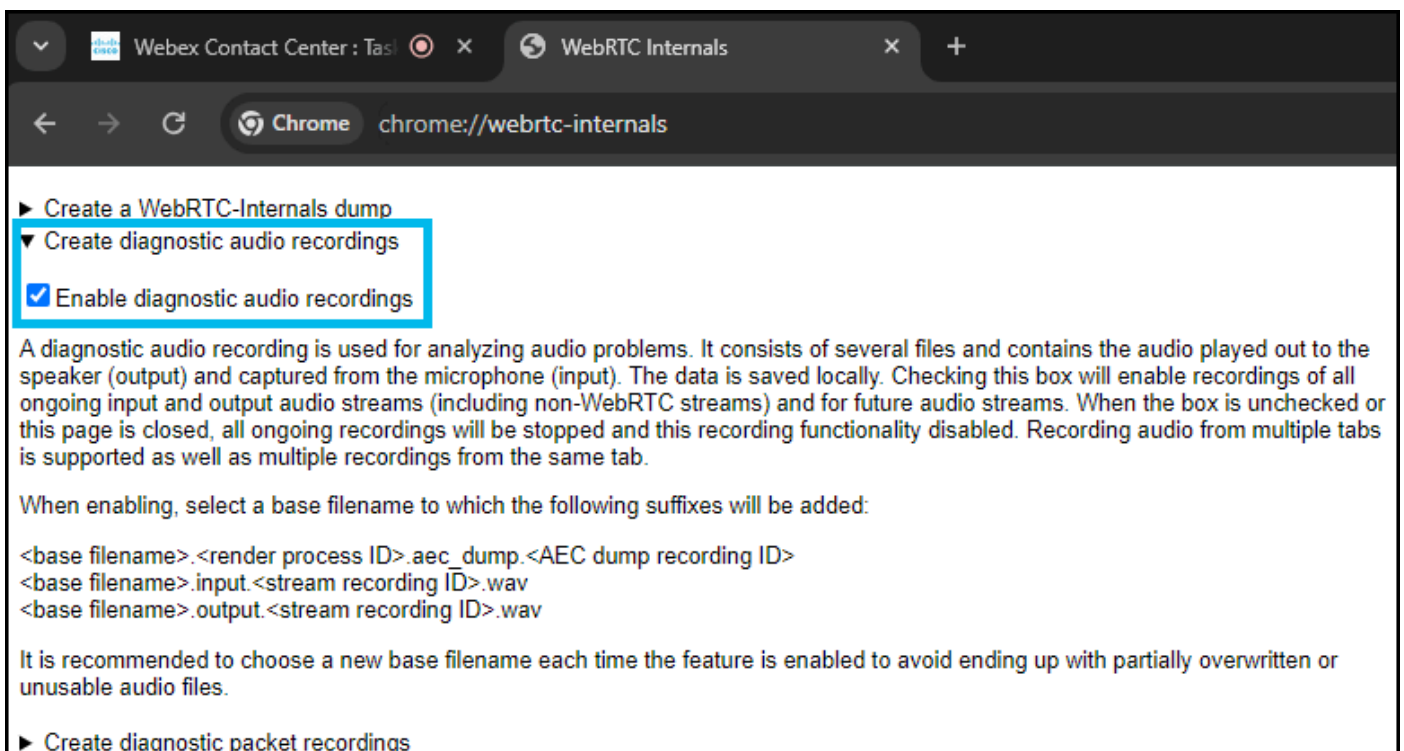


WebRTC-Protokoll-Dump kann durch Klicken auf 'Create a WebRTC-Internals Dump' und Klicken auf 'Download the "webrtc-internals Dump" Option abgerufen werden.



Anmerkung: Die WebRTC Log Dump-Datei kann mithilfe spezieller Tools analysiert und analysiert werden. <https://fippo.github.io/webrtc-dump-importer/>

WebRTC-Audio kann auch zu Diagnosezwecken aufgezeichnet werden.



Informationen zu dieser Übersetzung

Cisco hat dieses Dokument maschinell übersetzen und von einem menschlichen Übersetzer editieren und korrigieren lassen, um unseren Benutzern auf der ganzen Welt Support-Inhalte in ihrer eigenen Sprache zu bieten. Bitte beachten Sie, dass selbst die beste maschinelle Übersetzung nicht so genau ist wie eine von einem professionellen Übersetzer angefertigte. Cisco Systems, Inc. übernimmt keine Haftung für die Richtigkeit dieser Übersetzungen und empfiehlt, immer das englische Originaldokument (siehe bereitgestellter Link) heranzuziehen.