

WxCC Agent Desktop에서 WebRTC 데이터 및 통계 탐색

목차

[소개](#)

[사전 요구 사항](#)

[요구 사항](#)

[사용되는 구성 요소](#)

[개요](#)

[크롬](#)

소개

이 문서에서는 내장 브라우저 도구를 사용하여 사용 가능한 WebRTC 통화 데이터 및 통계에 대해 설명합니다.

사전 요구 사항

요구 사항

Cisco에서는 Webex Contact Center(WxCC) 솔루션, Chrome, Edge, Firefox 브라우저 및 WebRTC 프로젝트에 대한 기술 및 지식을 보유하고 있는 것이 좋습니다.

사용되는 구성 요소

이 문서는 특정 소프트웨어 및 하드웨어 버전으로 한정되지 않습니다.

이 문서의 정보는 특정 랩 환경의 디바이스를 토대로 작성되었습니다. 이 문서에 사용된 모든 디바이스는 초기화된(기본) 컨피그레이션으로 시작되었습니다. 현재 네트워크가 작동 중인 경우 모든 명령의 잠재적인 영향을 미리 숙지하시기 바랍니다.

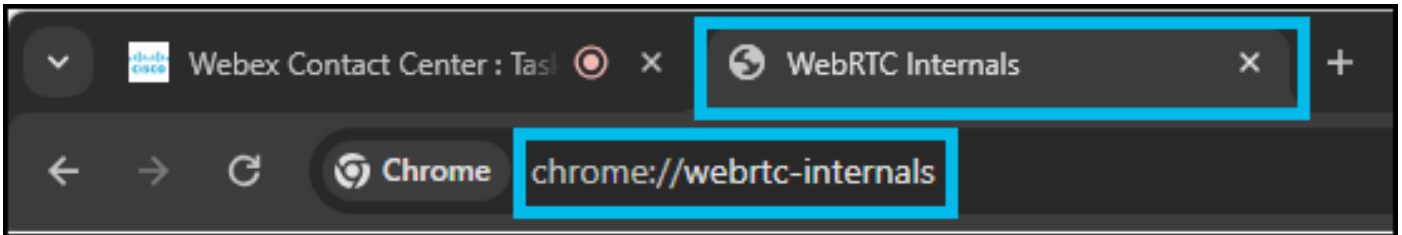
개요

WebRTC를 지원하는 대부분의 웹 브라우저에는 WebRTC 관련 문제를 해결하는 데 유용한 WebRTC 도구가 내장되어 있습니다.

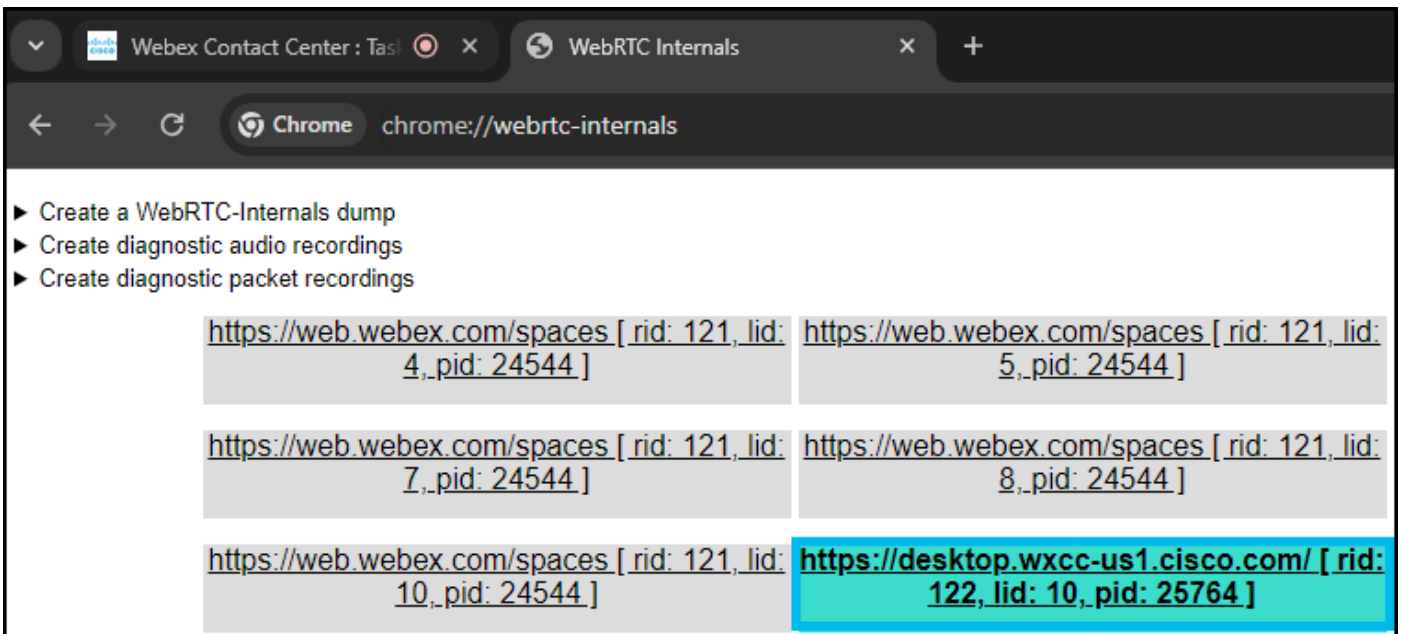
크롬

WebRTC 통화 데이터 및 통계에 액세스하려면 활성 통화 중에 Google Chrome 웹 브라우저에서 새 탭을 열어야 하며 다음 URL을 사용해야 합니다.

chrome://webrtc-internals



WebRTC-internals(WebRTC 내부) 탭에서 활성 WebRTC 세션을 선택해야 합니다(이름: desktop.wxcc-us1.cisco.com; URL은 WxCC 지역에 따라 다릅니다).



다음 정보를 볼 수 있습니다.

- 왼쪽(1)에서 PeerConnection API 활동 레코드를 사용할 수 있습니다. RTCPeerConnection은 WebRTC API의 중앙 인터페이스입니다. 로컬 피어와 원격 피어 간의 연결을 나타내며, 연결을 설정하는 데 필요한 모든 기능과 이벤트를 제공합니다(자세한 내용은 <https://webrtc.org/getting-started/overview>에서 확인할 수 있음). 이러한 추적(1)은 사용된 RTCPeerConnection 객체 및 매개 변수와의 상호 작용을 표시합니다.
- 오른쪽 (2)에서 현재 WebRTC 연결에 대해 getStats API를 통해 수집된 메트릭을 사용할 수 있습니다.
- 하단(3)에서는 현재 WebRTC 연결에 대한 getStats API(2)를 사용하여 생성한 그래프를 사용할 수 있습니다.

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 [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-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 [separate multiple values by " "]

- ▶ 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)

이 특정 WebRTC 연결에 사용 중인 로컬 및 원격 IP 및 포트는 PeerConnection API 호출의 추적 아래에서 볼 수 있습니다.

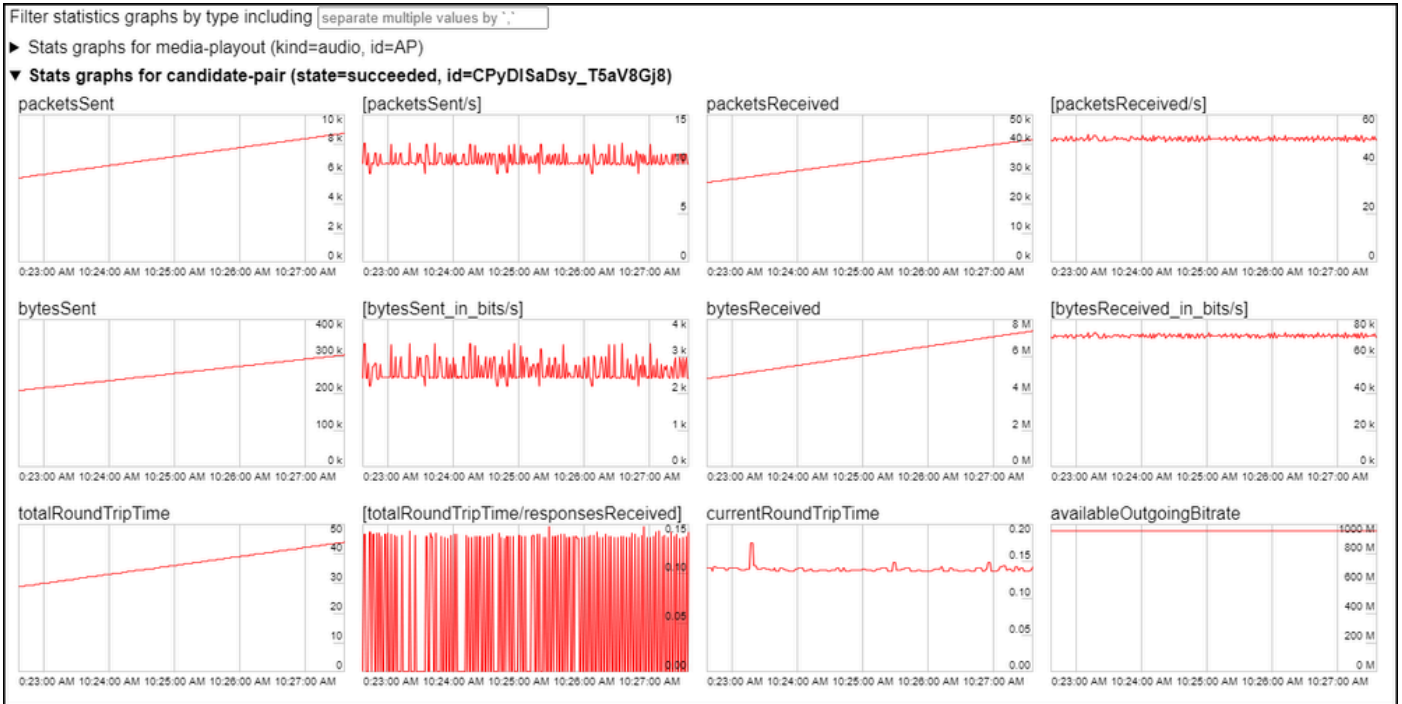
ICE connection state: new => completed
 Connection state: new => connected
 Signaling state: new => stable

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

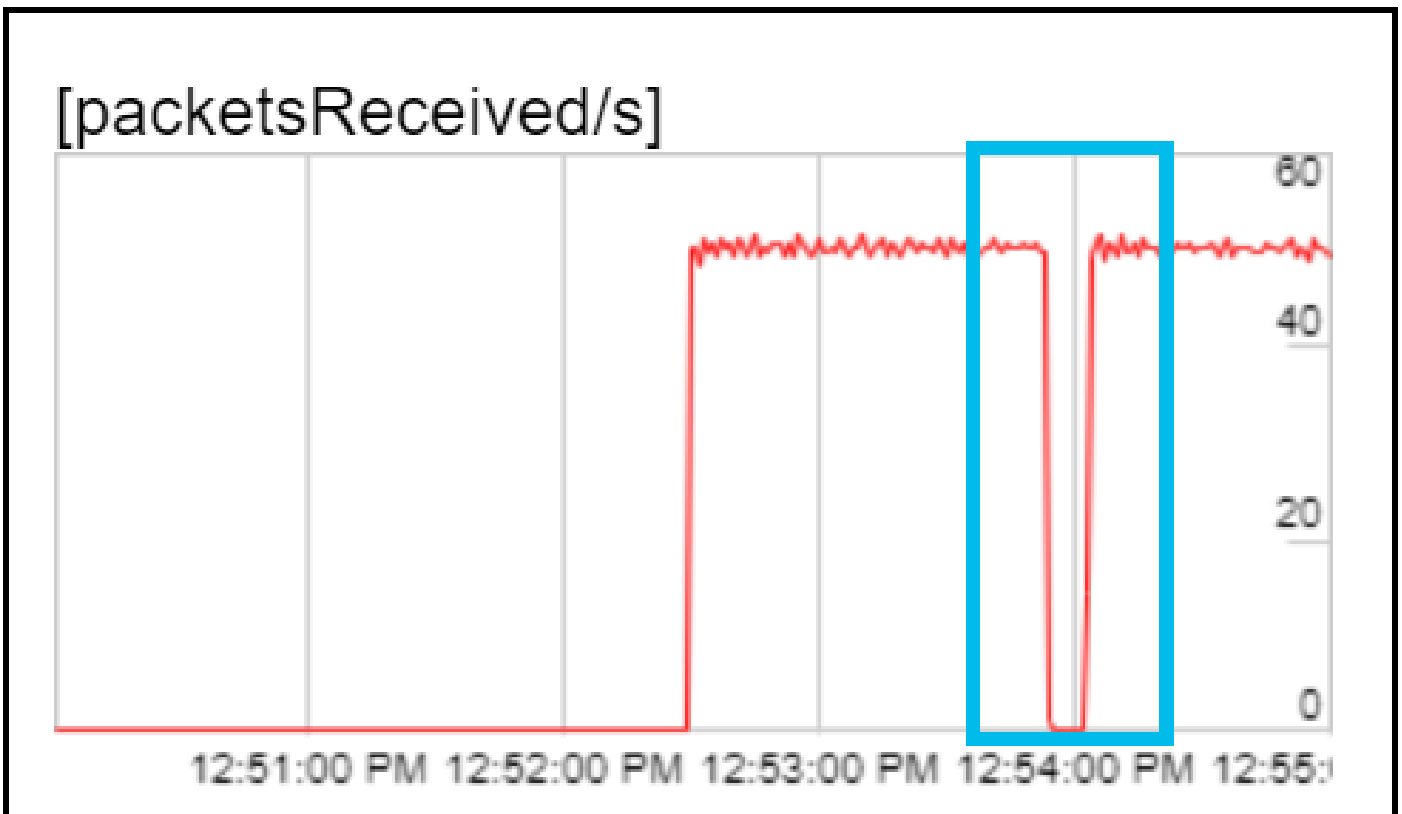
▶ ICE candidate grid

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

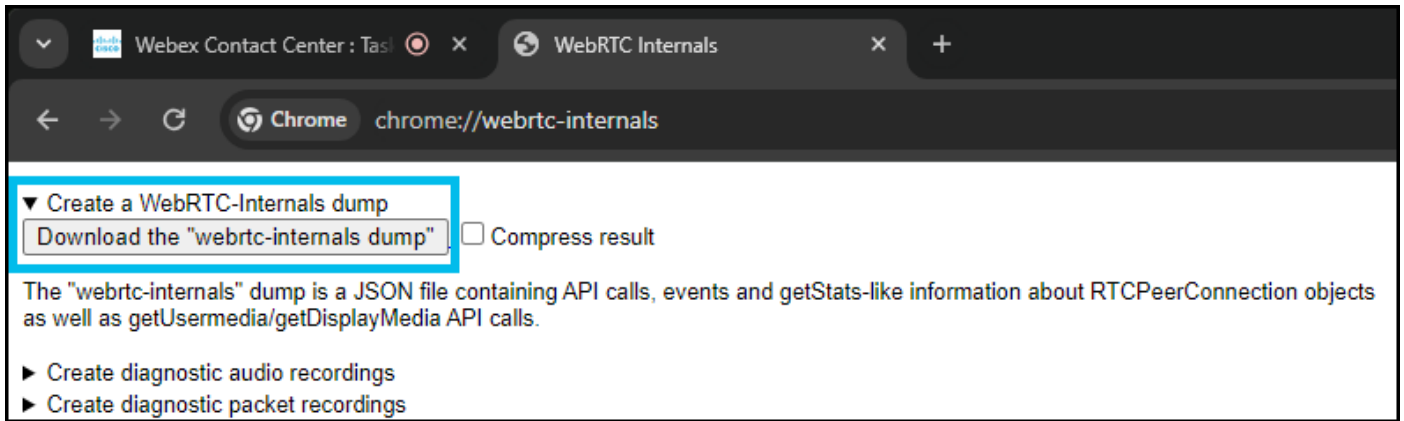
다양한 라이브 통화 통계(즉, 보낸 패킷, 받은 패킷, 대기 시간, 삭제된 패킷 등) 관련 그래프는 맨 아래에 표시됩니다(굵은 텍스트).



예를 들어, 수신 패킷이 누락되면 오디오가 누락됩니다.



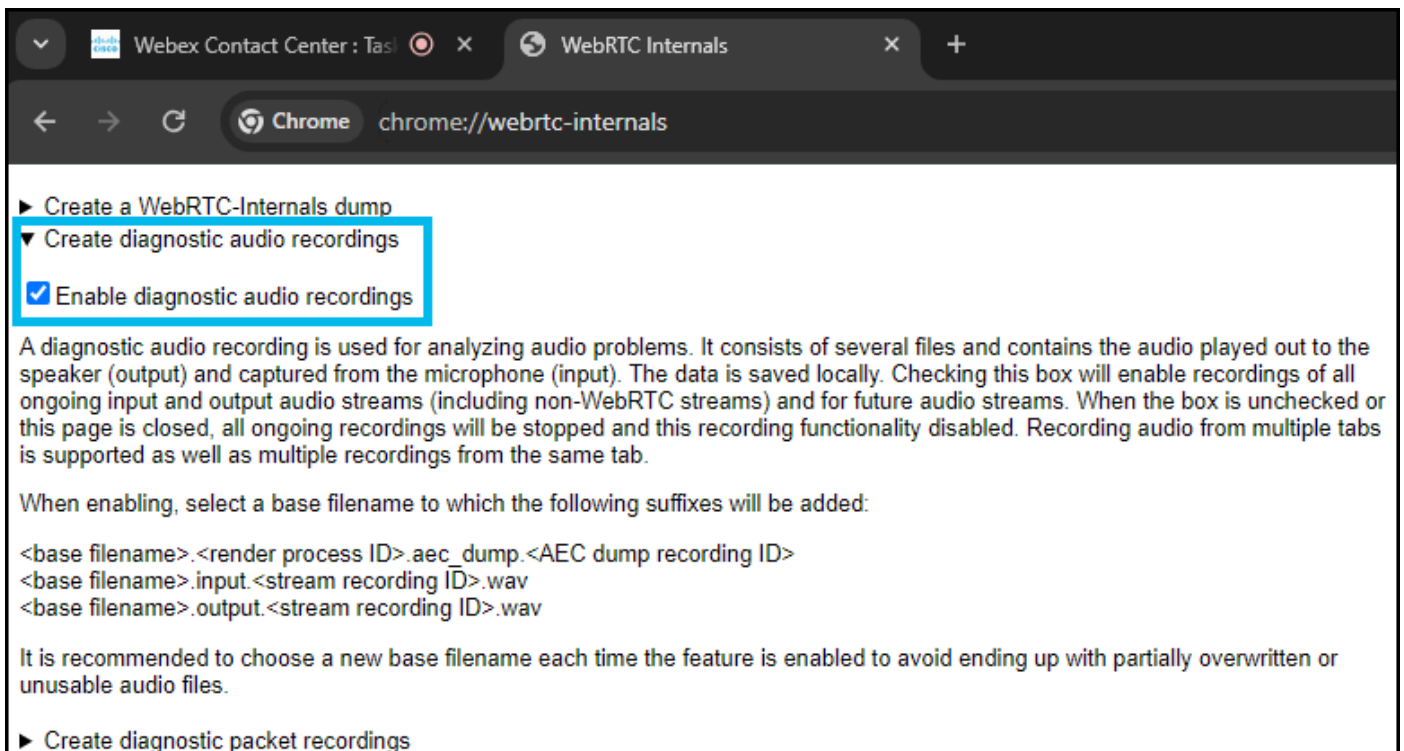
'WebRTC-Internals 덤프 만들기'를 클릭하고 'WebRTC-Internals 덤프 다운로드' 옵션을 클릭하여 WebRTC 로그 덤프를 검색할 수 있습니다.



참고: WebRTC 로그 덤프 파일은 특정 도구를 사용하여 구문 분석 및 분석할 수 있습니다.

<https://fippo.github.io/webrtc-dump-importer/>

WebRTC 오디오도 진단 목적으로 캡처할 수 있습니다.



이 번역에 관하여

Cisco는 전 세계 사용자에게 다양한 언어로 지원 콘텐츠를 제공하기 위해 기계 번역 기술과 수작업 번역을 병행하여 이 문서를 번역했습니다. 아무리 품질이 높은 기계 번역이라도 전문 번역가의 번역 결과물만큼 정확하지는 않습니다. Cisco Systems, Inc.는 이 같은 번역에 대해 어떠한 책임도 지지 않으며 항상 원본 영문 문서(링크 제공됨)를 참조할 것을 권장합니다.