CUBE(Unified Border Element) 및 TDM(Time-Division Multiplexing) 게이트웨이에 대한 디버그 컬렉션 구성

목차

소개

이 문서에서는 Cisco IOS/IOS-XE Voice Router에서 음성 디버그를 수집하기 위한 몇 가지 모범 사례에 대해 설명합니다.

사전 요구 사항

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.

요구 사항

- ISR(Integrated Services Router) 내부의 Cisco IOS/IOS-XE에 대한 기본 지식
- ISR 라우터에서 명령을 실행하기 위한 권한 부여된 액세스
- VoIP(Voice-over-IP) 프로토콜에 대한 이전 경험이 필요합니다.
- VoIP 추적의 경우 최소 Cisco IOS-XE 17.4.1 또는 17.3.2가 필요합니다.

사용되는 구성 요소

이 문서에서 사용되는 구성 요소는 다음과 같습니다.

- Cisco ISR 3925
- Cisco ISR 4451
- PuTTY

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.

배경

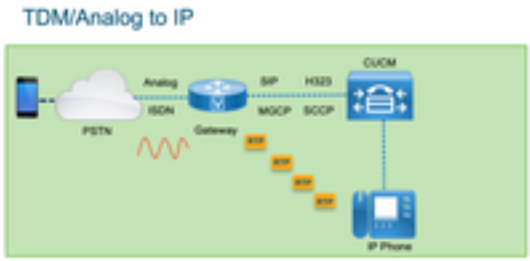
이러한 플랫폼의 디버그 수집 프로세스에는 문제가 있으며 디바이스의 성능에 영향을 미칠 수 있습니다. 음성 라우터에 여러 활성 통화가 설정된 경우 문제 및 위험이 증가합니다. 일부 시나리오에서 디버그가 올바르게 수집되지 않으면 CPU가 높아져 라우터의 용량이 손상되고 소프트웨어 충돌이 발생할 수 있습니다. 이 문서에서는 Cisco CUBE(Unified Border Element)와 TDM/아날로그 게이트

웨이의 차이점에 대해 설명합니다.

TDM 음성 게이트웨이 vs CUBE

TDM 음성 게이트웨이는 주로 내부 전화 시스템을 다른 PBX(Private Branch eXchange) 또는 PSTN(Public Switched Telephony Network)과 상호 연결하는 데 사용됩니다. TDM 게이트웨이에서 사용되는 연결 유형은 T1/E1 컨트롤러(ISDN 또는 CAS) 및 FXS 및 FXO 포트와 같은 아날로그 회로입니다. DSP(Digital Signal Processor)는 원시 형식의 오디오를 RTP 패킷으로 변환합니다. 유사한 방식으로, RTP 패킷은 DSP가 RTP 패킷을 처리하고 오디오를 특정 회로로 전송한 후 원시 오디오로 변환됩니다. 이러한 게이트웨이는 VoIP 측에서 H323, MGCP 또는 SCCP와 상호 운용될 수 있으며, TDM 측에서 ISDN PRI 회로 또는 아날로그 중 하나를 PSTN 또는 엔드포인트에 대한 가장 일반적인 연결로 사용할 수 있습니다.

그림에서 볼 수 있듯이 TDM 게이트웨이는 내부 VoIP 인프라와 아날로그 또는 ISDN 서비스 공급자 간에 브리지를 제공합니다.



VoIP의 도입으로 고객은 레거시 시스템을 최신 VoIP 인프라로 빠르게 변경하기 시작했습니다. 이제 연결을 사용하여 온프레미스 텔레포니 서비스를 통신 사업자 VoIP 인프라와 상호 연결하고 기능을 확장하여 더 나은 서비스를 제공하는 통신 사업자 측면에서도 마찬가지입니다. 오늘날 가장 많이 사용되는 VoIP 프로토콜은 SIP(Session Initiation Protocol)이며 현재 전 세계 고객과 ITSP(Internet Telephony Service Providers)에서 널리 사용되고 있습니다.

CUBE는 SIP를 기본 VoIP 프로토콜로 사용하는 ITSP를 통해 내부 VoIP 시스템을 외부 세계와 연결하는 방법을 제공하기 위해 도입되었습니다. CUBE는 단순히 T1/E1 컨트롤러나 아날로그 포트와 같은 TDM 유형의 연결이 더 이상 필요하지 않은 IP-IP 게이트웨이입니다. CUBE는 TDM 게이트웨이와 동일한 플랫폼에서 실행됩니다.

가장 일반적인 VoIP 프로토콜은 통화 설정 및 해제에 사용되는 SIP와 미디어 전송에 사용되는 RTP입니다. 트랜스코더가 필요하지 않은 경우 CUBE에서는 DSP가 필요하지 않습니다. RTP 트래픽은 ITSP에서 엔드포인트로 종단간 전송되며, CUBE는 주소가 숨겨져 있는 Middle-Man의 역할을 하며 제공되는 여러 기능 중 하나입니다.

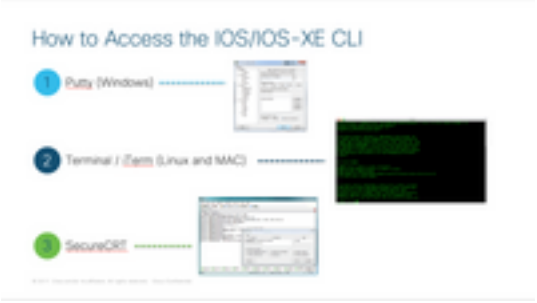
그림에서 볼 수 있듯이 CUBE는 내부 VoIP 인프라와 SIP ITSP를 나눕니다.

Cisco IOS/IOS-XE Voice Debug 컬렉션

음성 기능은 ISR, ASR, CAT8K와 같은 다양한 플랫폼에서 실행되지만 Cisco IOS 또는 Cisco IOS-XE인 공통 소프트웨어를 사용합니다(이 문서에서는 Cisco IOS와 Cisco IOS-XE의 차이점에 대해 다루지 않습니다). Cisco IOS 라우터에 액세스하는 방법에 대한 기본 사항부터 살펴보겠습니다.

CLI(Command Line Interface)를 통해 Cisco IOS/IOS-XE Router에 액세스하는 방법

다른 CLI 기반 디바이스와 마찬가지로 라우터도 SSH(Secure Shell) 또는 텔넷을 통해 명령을 실행하려면 터미널 모니터에 액세스해야 합니다. SSH는 디바이스에 대한 안전하고 암호화된 연결을 제공하므로 디바이스에 액세스하는 데 현재 사용되는 가장 일반적인 프로토콜입니다. 라우터의 CLI에 액세스하는 데 사용되는 일부 공통 터미널 모니터는 다음과 같습니다.

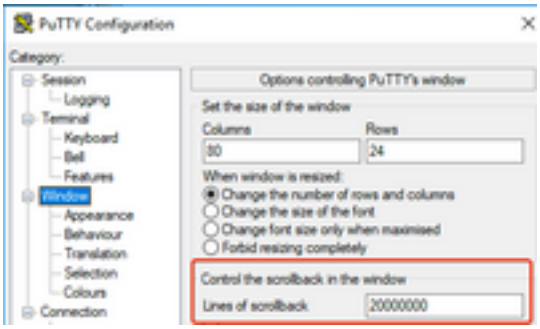


터미널 모니터를 show 명령 또는 디버그를 수집하도록 설정하는 방법

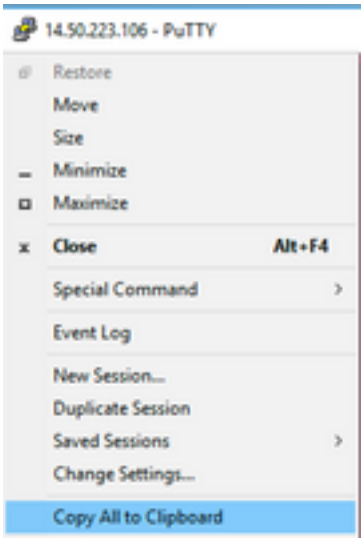
CLI에서 출력을 수집하는 방법에는 여러 가지가 있습니다. 라우터의 CLI에서 별도의 파일로 정보를 내보내는 것이 좋습니다. 이를 통해 외부 당사자에게 정보를 공유하는 것이 보다 용이하다.

디바이스에서 출력을 수집하는 두 가지 방법은 다음과 같습니다.

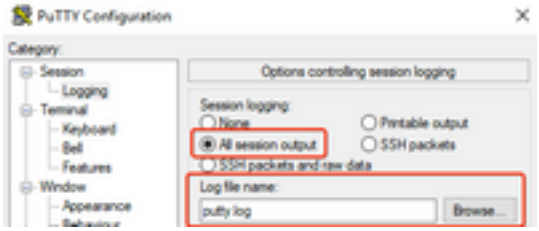
- 터미널의 모든 출력을 덤프합니다. 이 경우 scrollback 행이 충분한지 확인해야 합니다. 그렇지 않으면 scrollback이 출력의 첫 번째 섹션을 놓쳐 데이터가 불완전할 수 있습니다. Putty에서 scrollback 라인을 늘리려면 Putty Configuration(Putty 컨피그레이션) > Window(창) > Lines of Scrollback(Scrollback 라인)으로 이동합니다. 일반적으로 이 값은 scrollback 출력이 충분하도록 매우 높은 값으로 설정됩니다.



나중에 [클립보드에 **모두 복사**] 옵션을 사용하여 터미널 모니터에서 정보를 수집하고 출력을 텍스트 파일에 붙여넣을 수 있습니다.



- 또 다른 옵션은 전체 세션 출력을 .txt 파일에 로깅하는 것입니다. 이 옵션을 사용하면 입력한 모든 명령 및 수집된 출력이 텍스트 파일에 즉시 기록됩니다. 이는 세션의 모든 출력을 로깅하는 일반적인 방법입니다. 모든 세션 출력을 Putty의 파일에 로깅하려면 Putty Configuration(Putty 컨피그레이션) **> Session(세션) > Logging(로깅)**으로 이동한 다음 All Session Output(모든 세션 출력)을 다음과 같이 선택합니다.



> **참고:** 기본 로그 파일 이름은 다른 이름을 지정하지 않은 경우 사용됩니다. 나중에 찾기 위해 파일이 저장된 위치를 정확하게 확인하려면 찾아보기 버튼을 클릭하십시오. 또한 동일한 파일 경로에 있는 다른 putty.log 파일을 덮어쓰지 마십시오.

CLI에서 기본 show 명령 출력 수집

디버그 수집이 발생하기 전에 라우터에서 기본 정보를 수집하려면 Show 명령이 필요합니다. Show 명령은 빠른 속도로 수집되므로 대부분의 경우 라우터의 성능에 영향을 주지 않습니다. show 명령 출력만으로 즉시 문제가 격리될 수 있습니다.

라우터에 연결되면 터미널 길이를 0으로 설정할 수 있습니다. 이렇게 하면 모든 출력을 한 번에 표시하고 스페이스 바를 사용하지 않도록 수집이 빨라집니다. 라우터에 대한 자세한 정보를 수집하는 하나의 명령은 'show tech'이며, 또는 라우터에서 활성화된 음성 기능에 더 구체적인 데이터를 표시하는 **show tech voice**를 수집할 수 있습니다.

```
Router# terminal length 0
Router# show tech
!or
Router# show tech voice
Router# terminal default length !This cmd restores the terminal length to default
```

CLI에서 디버그 출력 수집

Cisco IOS/IOS-XE의 디버그 출력 수집은 때때로 문제가 될 수 있습니다. 라우터 충돌이 발생할 위

험이 있기 때문입니다. 하지만 문제를 방지하기 위해 다음 섹션에서 몇 가지 모범 사례에 대해 설명합니다.

메모리 검사

디버그를 활성화하기 전에 버퍼에 출력을 저장할 충분한 메모리가 있는지 확인해야 합니다.

show **process memory 명령**을 실행하여 모든 출력을 버퍼에 기록하기 위해 할당할 수 있는 메모리의 양을 확인합니다.

> **팁:** 터미널에 **표시되는** 제한된 **수의 줄로 돌아가려면 terminal length default 또**는 terminal length <num_lines> 명령을 사용합니다.

```
Router# show process memory
Processor Pool Total: 8122836952 Used: 456568400 Free: 7666268552
lsmpi_io Pool Total: 6295128 Used: 6294296 Free: 832
```

이 예에서는 라우터에서 사용할 수 있는 7666268552바이트(7.6GB)가 있습니다. 이 메모리는 모든 시스템 프로세스 간에 라우터에서 공유됩니다. 즉, 사용 가능한 전체 메모리를 사용하여 버퍼에 출력을 로깅할 수는 없지만 필요에 따라 충분한 시스템 메모리를 사용할 수 있습니다.

대부분의 시나리오에서는 출력을 손실하거나 덮어쓰기 전에 충분한 디버그 출력을 수집하기 위해 10MB 이상이 필요합니다. 드문 경우이지만 더 많은 양의 데이터를 수집해야 하는 경우, 이러한 특정 시나리오에서 50MB에서 100MB의 출력을 버퍼에 가져올 수 있습니다. 또는 사용 가능한 메모리가 있는 한 더 높게 이동할 수 있습니다.

사용 가능한 메모리가 부족하면 메모리 누수 문제가 발생할 수 있습니다. 그렇다면 아키텍처 TAC 팀과 함께 이러한 메모리 부족의 원인이 될 수 있는 사항을 수정하십시오.

CPU(Central Processing Unit) 확인

CPU는 시스템에서 활성화된 프로세스, 기능 및 통화의 양에 영향을 받습니다. 시스템에서 더 많은 기능 또는 통화가 활성화될수록 CPU가 더 바빠집니다.

좋은 벤치마크는 라우터의 CPU가 30% 이하인지 확인하는 것입니다. 즉, 기본 디버그에서 고급 디버그를 안전하게 활성화할 수 있습니다(고급 디버그가 사용될 때 항상 CPU를 주시해야 함). 라우터 CPU가 약 50%인 경우 기본 디버그를 실행하고 CPU를 주의 깊게 모니터링할 수 있습니다. CPU가 80% 이상이 되면 디버그를 즉시 중지하고(이 문서의 뒷부분에 표시됨) TAC에 지원을 요청하십시오.

정렬된 show process cpu 사용 **| 0.00 명령**을 **제외하여** 상위 프로세스와 함께 최근 5초, 60초 및 5분 CPU 값을 확인합니다.

```
Router# show processes cpu sorted | exclude 0.00
CPU utilization for five seconds: 1%/0%; one minute: 0%; five minutes: 0%
PID Runtime(ms) Invoked uSecs 5Sec 1Min 5Min TTY Process
211 4852758 228862580 21 0.15% 0.06% 0.07% 0 IPAM Manager
84 3410372 32046994 106 0.07% 0.04% 0.05% 0 IOSD ipc task
202 3856334 114790390 33 0.07% 0.05% 0.05% 0 VRRS Main thread
```

출력에서 라우터의 활동이 많지 않고 CPU가 낮으며 디버그를 안전하게 활성화할 수 있습니다.

주의: CPU가 50% 이상이고 최상위 프로세스가 음성 프로세스인 경우 기본 디버그만 활성화할 수 있도록 최상위 CPU 프로세스 활성 상태에 각별히 유의하십시오. 이 명령을 사용하여 CPU를 지속적으로 모니터링하여 라우터의 전체 성능에 영향을 미치지 않도록 합니다.

현재 진행 중인 통화 확인

각 라우터에는 서로 다른 용량 임계값이 있습니다. 최대 용량에 근접하지 않도록 라우터에서 활성화된 통화 수를 확인하는 것이 중요합니다. Cisco Unified Border Element 버전 12 데이터 시트에서는 참조할 수 있도록 각 플랫폼 용량에 대한 정보를 제공합니다.

show call **active total-calls** 명령을 사용하여 시스템에서 활성화된 통화 수를 파악할 수 있습니다.

```
Router# show call active total-calls
Total Number of Active Calls : 0
```

show call active voice summary 명령을 사용하여 활성화된 특정 통화 유형에 대한 자세한 정보를 가져올 수 있습니다.

```
Router# show call active voice summary
Telephony call-legs: 0
SIP call-legs: 0
H323 call-legs: 0
Call agent controlled call-legs: 0
SCCP call-legs: 0
STCAPP call-legs: 0
Multicast call-legs: 0
Total call-legs: 0
```

몇 가지 일반적인 값은 다음과 같습니다.

- **텔레포니 통화 레그**: TDM 게이트웨이 통화는 아날로그 및 PRI/ISDN 통화를 포함합니다.
- **SIP 호출 레그**: 총 SIP 통화 CUBE 라우터인 경우 통화당 2개의 통화 레그가 표시됩니다. 여기에 표시된 총 통화를 2로 나누어 정확한 번호를 얻습니다.
- **H323 호출 레그**: 총 H323 통화
- **SCCP 호출 레그**: 트랜스코더 및 MTP와 같은 라우터에 사용되는 CUCM 제어 미디어 리소스

로깅 버퍼 설정

라우터가 디버그 출력을 버퍼에 저장하도록 구성하려면 CLI의 설정을 수동으로 조정하도록 configure terminal 모드를 입력합니다. 이 컨피그레이션은 라우터에 영향을 미치지 않습니다. 그러나 이전 섹션에서 설명한 것처럼 컨피그레이션을 롤백해야 하는 경우 라우터에서 **show tech 또는 show running-config** 명령이 필요합니다.

구성 예는 TAC 엔지니어가 사용하는 공통 기준선입니다. 이 예에서는 10MB의 버퍼 메모리를 할당하지만 필요에 따라 늘릴 수 있습니다.

```
# configure terminal
service timestamps debug datetime msec localtime show-timezone year
service timestamps log datetime msec localtime show-timezone year
service sequence-numbers
logging buffered 10000000
no logging console
no logging monitor
```

```
logging queue-limit 10000
logging rate-limit 10000
voice iec syslog
```
명령은 다음 작업을 수행합니다.

- **서비스 타임스탬프 디버그 또는 로그**: 밀리초 정확도로 로컬 라우터 시간이 로깅된 모든 메시지에 기록되도록 합니다. 시간을 기준으로 통화를 찾는 데 유용합니다. 밀리초의 타임스탬프를 사용하면 두 행이 같은 밀리초 내에 발생할 때 디버그 행을 논리적 관련 이벤트로 그룹화할 수 있습니다.
- **서비스 시퀀스 번호:** 행에 디버그의 시퀀스 번호를 씁니다. 이는 로그가 syslog 서버로 전달될 때 유용합니다(필수). 이는 syslog 서버에 대한 디버그 메시지가 네트워크에서 삭제되었는지 식별하기 위해 매우 유용합니다. 시퀀스 번호는 타임스탬프 및 실제 로그 메시지 이전의 디버그 내 첫 번째 항목입니다. 이는 syslog 서버가 파일에 로컬로 쓸 수 있는 타임스탬프/시퀀스 번호와 다릅니다.
- **로깅 버퍼**: 로컬 버퍼 메모리로 디버그를 전송하도록 라우터에 알립니다. 버퍼 크기는 바이트 단위로 설정됩니다. 구성에서 버퍼 크기는 10MB로 설정되었습니다.
- **로깅 콘솔 없음** 및 **로깅 모니터 없음:** 콘솔 또는 터미널 모니터에 로그 메시지가 인쇄되지 않습니다. 이러한 명령이 구성되지 않으면 라우터 성능 및 디버그 출력 정확성에 문제가 될 수 있습니다.
- **음성 iec syslog:** 음성 내부 오류 코드 메시지를 활성화하여 연결 끊김 사유를 확인합니다.

Syslog 설정 구성

때로는 문제가 무작위로 발생할 수 있으며 이벤트가 발생할 때까지 디버그를 지속적으로 수집할 방법이 필요합니다. 디버그를 버퍼에 저장하면 지속적으로 수집합니다. 할당할 수 있는 메모리의 양으로 제한되며, 해당 메모리 양에 도달하면 버퍼가 가장 오래된 메시지를 둘러싸고 삭제하므로 문제를 격리하는 데 필요한 중요한 정보가 불완전하게 됩니다.

Syslog를 사용하면 라우터가 모든 디버그 메시지를 외부 서버로 전송할 수 있으며, Syslog 서버 소프트웨어는 이 메시지를 텍스트 파일에 저장합니다. 디버그 출력을 수집하는 좋은 방법이지만, 는 로그 수집에 선호되지 않습니다. Syslog 서버는 서버의 혼잡으로 인해 수신된 출력에서 라인을 건너뛰거나 삭제하는 경향이 있습니다. 디버그 출력이 서버를 압도하거나 네트워크 조건으로 인해 패킷이 삭제될 수 있기 때문입니다. 그러나 일부 시나리오에서는 Syslog가 문제를 진행하는 유일한 방법입니다.

가능하면 정보의 손실을 방지하기 위해 TCP와 같은 신뢰할 수 있는 전송 방법을 사용하고, 라우터가 연결되어 있는 동일한 스위치에 Syslog 서버를 연결하거나 라우터에 최대한 가깝게 연결해야 합니다. 모든 데이터가 파일에 저장되도록 보장하지는 않지만 데이터 손실 가능성을 줄입니다.

기본적으로 syslog 서버는 포트 514에서 전송 프로토콜로 UDP를 사용합니다.

```
#configure terminal
service timestamps debug datetime msec localtime show-timezone year
service timestamps log datetime msec localtime show-timezone year
service sequence-numbers

!Optional in case you still want to store debug output in the buffer.
logging buffered 10000000

no logging console
no logging monitor
```

```
logging trap debugging

!Replace the 192.168.1.2 with the actual Syslog Server IP Address
logging host 192.168.1.2 transport [tcp|udp] port
```

명령이 구성되는 즉시 라우터는 메시지를 Syslog 서버 IP 주소로 즉시 전달합니다.

디버그 컬렉션

디버그가 활성화되면, 문제가 재현되기 전에 버퍼를 지워야 합니다. 이 작업은 출력이 최대한 깨끗한지 확인하고 분석에 필요하지 않은 추가 데이터를 방지하기 위해 수행됩니다. clear log 명령을 **실행하면** 버퍼가 지워집니다. 라우터에서 활성화된 다른 통화가 있고 디버그가 활성화된 경우 출력이 즉시 버퍼에 인쇄됩니다.

```
Router# clear log
Clear logging buffer [confirm]
Router#
```

문제가 재현되면 디버그를 즉시 비활성화하여 버퍼의 추가 출력을 중지합니다. 그런 다음 로그를 수집합니다. 다음 명령을 사용하여 터미널의 모든 출력을 덤프할 수 있습니다.

```
Router# undebug all
Router# terminal length 0
Router# show log
```

PuTTY는 한 번에 모든 출력을 처리하지 않기 때문에 닫히는 경우가 있습니다. 이는 정상이며 장애가 발생했다는 의미는 아닙니다. 이 경우 세션을 다시 열고 정상적으로 계속합니다. 로깅 버퍼가 너무 크거나 인쇄해야 하는 데이터 양으로 인해 터미널 모니터가 충돌하는 경우 명령 **show log**를 사용하여 버퍼 출력을 외부 장치에 직접 복사합니다 **| 리디렉션**:

```
Router# show log | redirect ftp://username:password@192.168.1.2/debugs.txt
```

이 명령은 전체 버퍼 출력을 파일 이름 debug.txt를 사용하여 IP 주소 192.168.1.2의 ftp에 복사합니다. 파일 이름은 항상 지정해야 합니다. 해당 데이터를 내보낼 수 있는 다른 대상은 다음과 같습니다.

```
Router# sh log | redirect ?
bootflash: Uniform Resource Locator
flash: Uniform Resource Locator
ftp: Uniform Resource Locator
harddisk: Uniform Resource Locator
http: Uniform Resource Locator
https: Uniform Resource Locator
nvram: Uniform Resource Locator
tftp: Uniform Resource Locator
```

음성 라우터에서 어떤 디버그를 활성화할 수 있습니까?

각 통화 흐름과 기능 유형(TDM, CUBE 또는 SCCP(미디어 리소스))이 다르며 활성화할 수 있는 특정 디버그가 있습니다. 필요한 모든 디버그를 동시에 활성화해야 합니다. 한 번에 하나의 디버그만 캡처하면 효과가 없으며 데이터를 분석할 때 더 많은 혼란을 제공합니다.

디버그는 CLI EXEC 프롬프트 레벨 Router#에서 활성화되어 특권 실행 모드 권한이 있어야 합니다.

기본 및 고급 디버그가 있습니다. 기본 디버그는 SIP, H323 또는 MGCP에서 신호 정보를 수집하는 데 사용되며, 이는 라우터가 피어 디바이스와 가진 대화를 보여줍니다.

고급 디버깅은 매우 상세하며, 일반적으로 기본 디버깅에서 표시할 수 없는 내부 스택 오류가 발생할 경우 추가 정보를 수집하는 데 사용됩니다. 이러한 디버깅은 일반적으로 CPU를 많이 사용합니다.

> **팁:** 디버그가 활성화된 후에는 clear logging 명령을 **실행해야 합니다**. 이 명령을 사용하면 디버그를 보다 깔끔하게 캡처할 수 있도록 버퍼를 지웁니다.

CCAPI(내부 통화 제어 API) 디버그

각 Cisco IOS/IOS-XE Router 내에는 서로 다른 VoIP 애플리케이션 또는 프로토콜과 데이터 플레인 구성 요소(예: RTP, DSP, 음성 카드) 간의 통신을 담당하는 통화 제어 API가 있습니다. 이 레이어에서 데이터를 캡처하기 위해 사용할 수 있는 특정 디버그가 있습니다.

```
debug voip ccapi inout
```

이 디버그에 대한 다른 옵션이 있지만 debug voip ccapi 입력은 일반적으로 이 레이어의 상태를 이해하는 데 충분한 것보다 많은 모든 기본 다이얼 플랜 및 통화 설정 정보를 포함합니다.

> **팁: debug voip ccapi inout**은 일반적으로 라우터의 CPU에 최소한의 영향을 미치며, 통화의 정보 및 여러 상태에 대한 전체 로그 집합을 제공하기 위해 신호 디버깅과 함께 활성화하는 것이 좋습니다.

SIP 통화 흐름

이러한 디버깅은 SIP 통화 흐름에 가장 일반적으로 사용되며 라우터와 CUCM 또는 다른 SIP 서버/프록시 간에 SIP 레그를 사용하여 CUBE 및 TDM 게이트웨이 내에서 활성화할 수 있습니다.

기본 SIP 디버그

```
debug ccsip messages
debug ccsip error
debug ccsip non-call !Optional, applies for SIP OPTIONS and SIP REGISTER Messages.
```

고급 SIP 디버깅

```
debug ccsip all
debug ccsip verbose
debug voice ccapi inout
```

디지털(PRI, BRI) 통화 흐름

이러한 디버깅은 PRI(Primary Rate Interfaces) T1/E1 또는 BRI(Basic Rate Interfaces)에 적용됩니다.

기본 디지털 디버그

```
debug isdn q931
```

고급 디지털 디버그

```
debug isdn q921
```

아날로그 통화 흐름

이러한 디버그는 FXS(Forehigh eXchange Subscriber) 또는 FXO(Foreign eXchange Office) 포트와 같은 아날로그 회로가 포함된 경우 사용됩니다.

```
debug vpm signal
debug voip vtsp all
```

MGCP 통화 흐름

이러한 디버그는 MGCP가 음성 게이트웨이와 CUCM 간의 음성 프로토콜로 사용될 때 사용됩니다.

기본 디버그

```
debug mgcp packets
debug mgcp errors
```

CCM-Manager 디버깅

디버그 ccm-manager는 CUCM과 음성 게이트웨이 간의 컨피그레이션 다운로드, MoH 및 PRI/BRI 백홀 메시지를 추적하는 데 사용됩니다. 이러한 디버깅은 필요에 따라 사용되며 오류 시나리오에 따라 달라집니다.

```
debug ccm-manager backhaul !For PRI and BRI Deployments
debug ccm-manager errors
debug ccm-manager events
debug ccm-manager config-download !Troubleshoot Configuration download issues from CUCM TFTP
debug ccm-mananger music-on-hold !Troubleshoot internal MoH Process
```

고급 MGCP 디버깅

```
debug mgcp all
```

H323 통화 흐름

H323이 널리 사용되고 있지는 않지만, H323이 구성된 일부 구축은 여전히 있습니다.

기본 H323 디버그

```
debug h225 asn1
debug h245 asn1
```

```
debug h225 events
debug h245 events
```

고급 H323 디버깅

```
debug cch323 h225
debug cch323 h245
debug cch323 all
```

SCCP 미디어 리소스

이러한 디버그는 MTP(Media Termination Point) 또는 CUCM(Cisco Unified Communications Manager) 서버에 등록된 트랜스코더와 관련된 SCCP(Skinny Call Control Protocol) 미디어 리소스 문제를 해결하는 데 사용됩니다.

기본 SCCP 디버깅

```
debug sccp messages
debug sccp events
debug sccp errors
```

고급 SCCP 디버그

```
debug sccp all
```

VoIP 추적

Cisco IOS-XE 17.4.1 및 17.3.2의 도입으로 CUBE(Cisco Unified Border Element) 내에서 음성 로그를 캡처하는 새로운 옵션이 추가되었습니다. 이 새로운 기능을 VoIP 추적이라고 합니다. 이는 디버그를 활성화할 필요 없이 SIP 신호 및 이벤트를 로깅할 수 있도록 생성된 새로운 서비스 가용성 프레임워크입니다.

VoIP 추적은 기본적으로 활성화되어 있으며 필요에 따라 언제든지 비활성화할 수 있습니다. VoIP 추적은 SIP 통화에 대해서만 특정 정보를 캡처합니다.

- SIP 트렁크 간 통화에 대한 SIP 메시지
- SIP 레이어에서 CUBE의 다른 레이어로의 이벤트 및 API 호출
- SIP 오류
- 통화 제어(CUBE에서 처리하는 통합 커뮤니케이션 통화 흐름)
- FSM(Finite State Machine) 상태 및 이벤트
- 다이얼 피어가 일치함
- 할당된 RTP 포트
- SIP 시그널링과의 IEC 오류 상관관계

제한 사항

- VoIP 추적은 Out-of-Dialog SIP 메시지와 관련된 정보를 기록하지 않습니다. 등록옵션구독/알림정보
- HA의 VoIP 추적은 지원되지만 다음 주의 사항이 적용됩니다. 스탠바이 라우터에는 기본적으로 VoIP 추적이 활성화되어 있습니다. 스탠바이 프로세스가 활성화될 때까지 해당 추적만 표시됩니다Standby가 Active이면 체크포인트 **통화**의 전체 추적은 포함되지 않으며 새 통화만 포함됩

니다**show voip trace <key>는** 대기 라우터에서 계속 작동하며 통화에 대한 커버 버퍼 및 미디어 스트림 데이터를 표시합니다

VoIP 추적을 활성화하는 방법

앞서 언급한 대로 이 기능은 기본적으로 활성화되어 있습니다. 이 기능을 활성화하는 명령은 다음과 같습니다.

```
Router# configuration terminal
Router(config)# voice service voip
Router(conf-voi-serv)# trace
Router(conf-serv-trace)#
```

VoIP 추적을 비활성화하는 방법

이 기능을 비활성화하려면 다음 명령을 사용하십시오.

```
Router(conf-serv-trace)# no trace
!or
Router(conf-serv-trace)# shutdown
```

주의: VoIP 추적이 비활성화되면 모든 메모리가 지워지고 정보가 손실됩니다.

추적 컨피그레이션 모드에서 사용할 수 있는 명령은 다음과 같습니다.

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

메모리 제한 구성

memory-limit은 VoIP 추적이 데이터를 저장하는 데 사용하는 메모리의 양을 결정합니다. 기본적으로 플랫폼에서 사용 가능한 메모리의 10%이지만, 최대 1GB, 최소 10MB로 변경할 수 있습니다. 동적으로 할당되는 메모리입니다. 즉, 이 기능은 필요에 따라 메모리만 사용하며 통화 볼륨에 종속됩니다. 사용 가능한 최대 메모리에 도달하면 이전 항목을 순환하여 삭제합니다.

메모리 제한이 사용 가능한 10% 메모리보다 크도록 수정되면 명령줄 인터페이스에 다음과 같은 메시지가 표시됩니다.

```
Router(conf-serv-trace)# memory-limit 1000
Warning: Setting memory limit more than 10% of available platform memory (166 MB) will affect
system performance.
```

기본값을 10% 메모리 사용량으로 설정하려면 다음과 같이 **memory-limit 플랫폼을** 사용할 수 있습니다.

```
Router(conf-serv-trace)# memory-limit platform
 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. Continue? [no]:
```

주의: 메모리 제한이 감소하면 모든 VoIP 추적 데이터가 손실됩니다. 메모리가 축소되기 전에 데이터의 백업을 수집해야 합니다.

VoIP 추적 데이터를 표시하는 방법

VoIP 추적의 데이터를 표시하려면 특정 show 명령을 사용해야 합니다. 데이터는 동일한 터미널 세션에 표시되거나 Syslog를 통해 오프박스 Syslog 서버로 전송될 수도 있습니다.

참고: 통화에 대한 BYE가 수신된 시간으로부터 32초 후에 추적이 덤프됩니다.

참고: SIP Signaling(SIP 신호 처리)은 레그별로 표시되며 일반 디버깅처럼 결합되지 않습니다. debug ccsip messages와 **같은 일반** 디버그는 이벤트가 발생한 정확한 순서대로 통화의 SIP 신호 처리를 표시합니다. VoIP 추적에서는 각 레그가 별개입니다. 올바른 순서를 결정하기 위해 타임스탬프가 사용됩니다.

데이터를 표시하는 데 사용할 수 있는 명령은 다음과 같습니다.

```
Router# show voip trace ?
 all            Display all VoIP Traces
 call-id        Filter traces based on Internal Call Id
 correlator     Filter traces based on FPI Correlator
 cover-buffers  Display the summary of all cover buffers
 session-id     Filter traces based on SIP Session ID
 sip-call-id    Filter traces based on SIP Call Id
 statistics     Display statistics for VoIP Trace
```

show voip trace all

이 명령은 버퍼에서 사용 가능한 모든 VoIP 추적 데이터를 표시합니다. 이 명령을 사용하면 라우터의 성능이 영향을 받습니다. 명령을 입력하면 위험에 대해 경고하고 계속할지 확인하는 경고 메시지가 표시됩니다.

```
Router# show voip trace all
 Displaying 11858 cover buffers
 This may severely impact system performance.
 Continue? [yes/no] no
```

show voip trace cover-buffers

이 명령은 VoIP Trace(VoIP 추적) 아래에서 보고된 모든 통화에 대한 통화 세부사항의 개요를 표시합니다. 각 통화 레그에는 기록된 통화의 요약을 포함하는 커버 버퍼가 생성됩니다.

```
Router# show voip trace cover-buffers
------------------ Cover Buffer ---------------
Search-key = 8845:3002:659
Timestamp = *Sep 30 01:17:33.615
Buffer-Id = 1
CallID = 659
Peer-CallID = 661
Correlator = 4
```

```
Called-Number = 3002
Calling-Number = 8845
SIP CallID = 20857880-1ec12085-13b930-411b300a@10.48.27.65
SIP Session ID = 2b1289c400105000a0002c3ecf872659
GUID = 208578800000
-----------------------------------------

------------------ Cover Buffer ---------------
Search-key = 8845:3002:661
Timestamp = *Sep 30 01:17:33.634
Buffer-Id = 2
CallID = 661
Peer-CallID = 659
Correlator = 4
Called-Number = 3002
Calling-Number = 8845
SIP CallID = 8D6DEC28-1F111EB-829FD797-1B22F6DB@10.48.55.11
SIP Session ID = 0927767800105000a0005006ab805584
GUID = 208578800000
-----------------------------------------
```

각 필드에 대한 자세한 내용은 다음 표를 참조하십시오.

필드	설명
검색 키	발신 번호 및 통화 ID의 조합을 포함합니다.
타임스탬프	커버 버퍼 생성 시간
버퍼 ID	커버 버퍼의 버퍼 ID
통화 ID	커버 버퍼에 대한 각 통화 레그의 통화 ID
피어 통화 ID	피어 레그의 통화 ID
상관기	통화의 FPI 상관기
수신 번호	커버 버퍼의 각 통화 레그의 수신 번호
호출 번호	커버 버퍼의 각 통화 레그의 발신 번호
Sip 통화 ID	커버 버퍼의 각 통화 레그의 SIP 통화 ID
Sip 세션 ID	커버 버퍼의 각 통화 레그의 SIP 세션 ID
GUID	커버 버퍼의 각 호출의 GUID입니다.
앵커 레그	각 통화 레그가 통화 포킹 흐름 또는 미디어 프록시 구축의 앵커 레그인 경우 앵커 레그는 예로 설정됩니다
갈라진 다리	각각의 통화 레그가 통화 포킹 흐름 또는 미디어 프록시 구축의 앵커 레그인 경우 포크된 레그는 예로 설정됩니다
연결된 Call ID	연결된 분기된 다리의 통화 ID

커버 버퍼를 필터링하기 위해 include 및 **section** 명령을 사용할 수 있습니다.

```
Router# show voip trace cover-buffers | include Search-key | 8845 | 3002
Search-key = 8845:3002:661
!or
Router# show voip trace cover-buffers | section Search-key | 8845 | 3002
Search-key = 8845:3002:661
```

show voip trace call-id

이전 명령과 함께 **show voip trace call-id**를 사용하여 통화를 찾을 수 있습니다. call-id가 식별되면 이 명령을 사용하여 특정 통화 레그에 대한 모든 정보를 표시할 수 있습니다.

```
Router# show voip trace cover-buffers | include Search-key | 8845 | 3002
```

```
Search-key = 8845:3002:661
Router# show voip trace call-id 661
```

voip 추적 통계 표시

이 show 명령은 상태, 메모리 소비, 오류 또는 실패 통화, 성공한 통화, 최신 및 오래된 항목의 타임스탬프 등에 대한 자세한 출력을 표시합니다.

```
Router# show voip trace statistics
VoIP Trace Statistics
Tracing status             : ENABLED at *Sep 12 06:44:02.349
Memory limit configured    : 803209216 bytes
Memory consumed            : 254550928 bytes (31%)
Total call legs dumped     : 2
Oldest trace dumped        : *Sep 12 07:29:21.077 Search-key: 9898:30000:64
Latest trace dumped        : *Sep 12 07:29:21.010 Search-key: 9898:30000:63
Total call legs captured   : 11858
Total call legs available  : 11858
Oldest trace available     : *Sep 12 06:57:23.923, Search-key: 5250001:4720001:11
Latest trace available     : *Sep 13 05:08:25.353, Search-key: 19074502232:30000:13177
Total traces missed        : 0
```

각 필드에 대한 자세한 내용은 다음 표를 참조하십시오.

필드	설명
추적 상태	VoIP 추적이 활성화된 시간 및 날짜를 포함하여 추적 상태를 표시합니다.
구성된 메모리 제한	구성된 메모리 제한을 표시합니다. 이는 프로세서 풀 메모리 크기의 10%입니다
사용된 메모리	VoIP 추적을 위해 동적으로 사용되는 메모리 양을 표시합니다.
덤프된 총 통화 레그	로깅 버퍼에 덤프된 실패한 통화 레그 수를 표시합니다. 덤프된 통화는 IEC 오류와 연결된 통화 레그를 가리킵니다.
가장 오래된 추적 덤프	VoIP 추적이 활성화된 이후 가장 오래된 실패한 통화의 타임스탬프 및 검색 키를 표시합니다
최신 추적 덤프	VoIP 추적이 활성화된 이후 실패한 최근 통화의 타임스탬프 및 검색 키를 표시합니다.
캡처된 총 통화 레그	VoIP 추적이 활성화된 후 캡처된 총 레그를 표시합니다
사용 가능한 총 통화 레그	기록에서 사용 가능한 총 통화 레그를 표시합니다. 이는 메모리 제한에 따라 캡처된 총 레그와 동일하거나 다를 수 있습니다.
사용 가능한 가장 오래된 추적	메모리에서 사용 가능한 가장 오래된 커버 버퍼의 타임스탬프 및 검색 키를 표시합니다
최신 추적 사용 가능	메모리에서 사용 가능한 최신 커버 버퍼의 타임스탬프 및 검색 키를 표시합니다
누락된 총 추적 수	메모리 제한으로 인해 누락된 통화 레그의 수를 표시합니다.

추가 show 명령

필드	사용	
show voip trace correlator <correlator>	show voip trace correlator 4	커버 버퍼에서 시작하는 특정 통화 ID에 대한 VOIP 추적을 필
show voip trace session-id <session-id>	show voip trace session-id 87003120822b5dbd8fd80f62d8e57c48	SIP 세션 ID를 기반으로 통화에 대한 V 더의 로컬 또는 원격 UUID를 사용하여
show voip trace sip-call-id <call-id>	show voip trace sip-call-id 01e60dfa9d8442848336d79e3155a8a1	SIP Call-ID를 기반으로 VOIP 추적을

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.