Konfigurieren der Debugsammlung für Unified Border Element (CUBE)- und Time Division Multiplexing (TDM)-Gateways

Inhalt

Einleitung

In diesem Dokument werden einige der Best Practices für die Erfassung von Sprachdebugs auf einem Cisco IOS/IOS-XE Voice Router beschrieben.

Voraussetzungen

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 verstehen.

Anforderungen

- Grundkenntnisse von Cisco IOS/IOS-XE in Integrated Services Routern (ISR)
- Privilegierter Zugriff, um Befehle in den ISR-Routern auszuführen.
- Eine umfassende Erfahrung mit Voice-over-IP (VoIP)-Protokollen ist wünschenswert.
- Für VoIP Trace ist mindestens Cisco IOS-XE 17.4.1 oder 17.3.2 erforderlich.

Verwendete Komponenten

Für dieses Dokument werden folgende Komponenten verwendet:

- Cisco ISR 3925
- Cisco ISR 4451
- PuTTY

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 verstehen.

Hintergrund

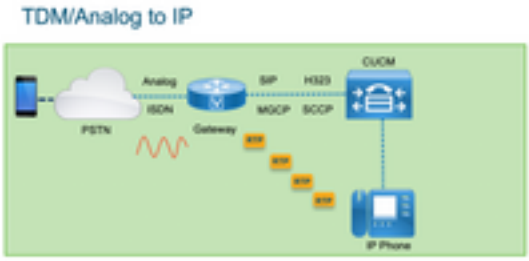
Der Prozess der Debug-Sammlung in diesen Plattformen hat Herausforderungen und könnte sich potenziell auf die Leistung des Geräts auswirken. Die Herausforderungen und Risiken nehmen zu,

wenn ein Voice Router mehrere aktive Anrufe umfasst. In einigen Szenarien, wenn die Debug-Meldungen nicht korrekt erfasst werden, kann dies zu einer hohen CPU führen, die die Kapazität des Routers beeinträchtigen und sogar einen Software-Absturz verursachen kann. In diesem Dokument wird der Unterschied zwischen einem Cisco Unified Border Element (CUBE) und einem TDM-/analogen Gateway erläutert.

TDM-Sprach-Gateways und CUBE

TDM-Sprach-Gateways werden hauptsächlich verwendet, um ein internes Telefonsystem mit einer anderen Telefonanlage (PBX) oder dem öffentlichen Telefonnetz (PSTN) zu verbinden. Die Verbindungen, die in TDM-Gateways verwendet werden, sind T1/E1-Controller (ISDN oder CAS) und analoge Schaltkreise wie FXS- und FXO-Ports. Ein Digital Signal Processor (DSP) wandelt die Audiodaten aus der Rohform in RTP-Pakete um. Auf ähnliche Weise werden RTP-Pakete in Raw-Audio umgewandelt, nachdem der DSP die RTP-Pakete verarbeitet hat und das Audio über den jeweiligen Schaltkreis sendet. Diese Gateways können mit H323, MGCP oder SCCP auf der VoIP-Seite und auf der TDM-Seite entweder seine ISDN PRI-Schaltungen oder Analog als die häufigsten Verbindungen zum PSTN oder Endpunkten zusammenarbeiten.

Wie im Bild gezeigt, stellen die TDM-Gateways eine Brücke zwischen Ihrer internen VoIP-Infrastruktur und den analogen oder ISDN-Service Providern dar.



Mit der Einführung von VoIP haben Kunden schnell damit begonnen, ihre alten Systeme in eine moderne VoIP-Infrastruktur umzuwandeln. Das Gleiche passierte auf der Seite der Service Provider, wo sie nun Verbindungen nutzen, um standortbasierte Telefoniedienste mit der VoIP-Infrastruktur des Service Providers zu verbinden und ihre Funktionen zu erweitern, um bessere Dienste bereitzustellen. Das heute am häufigsten verwendete VoIP-Protokoll ist das Session Initiation Protocol (SIP), das derzeit von Kunden und Internet-Telefonie-Service Providern (ITSP) weltweit verwendet wird.

CUBE wurde eingeführt, um diese internen VoIP-Systeme über die ITSPs mit SIP als primärem VoIP-Protokoll mit der Außenwelt zu verbinden. CUBE ist ein einfaches IP-IP-Gateway, für das keine TDM-Verbindungen wie T1/E1-Controller oder analoge Ports mehr erforderlich sind. CUBE wird auf denselben Plattformen wie TDM-Gateways ausgeführt.

Das am häufigsten verwendete VoIP-Protokoll ist SIP für die Herstellung und Beendigung von Anrufen und RTP für die Medienübertragung. In CUBE ist ein DSP nur dann erforderlich, wenn ein Transcoder erforderlich ist. Der RTP-Datenverkehr verläuft durchgängig vom ITSP zum Endpunkt. CUBE fungiert dabei als Vermittler, wobei Adressen als eine der vielen Funktionen ausgeblendet werden.

Wie im Bild gezeigt, ermöglicht CUBE eine Trennung zwischen Ihrer internen VoIP-Infrastruktur und dem SIP ITSP:

Sammlung von Cisco IOS/IOS-XE-Sprachdebugs

Sprachfunktionen werden auf einer anderen Liste von Plattformen ausgeführt, wie z. B. ISR, ASRs, CAT8Ks. Sie verwenden jedoch eine gemeinsame Software, die entweder Cisco IOS oder Cisco IOS-XE ist (die Unterschiede zwischen Cisco IOS und Cisco IOS-XE werden in diesem Artikel nicht behandelt). Beginnen wir mit den Grundlagen für den Zugriff auf den Cisco IOS Router.

Zugriff auf einen Cisco IOS/IOS-XE-Router über eine Kommandozeile (CLI)

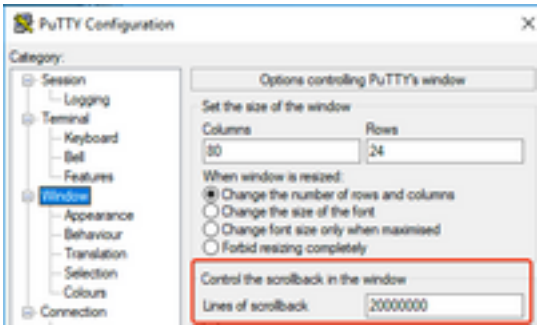
Wie alle anderen CLI-basierten Geräte benötigen Router einen Terminalmonitor, um über Secure Shell (SSH) oder Telnet auf die Befehle zugreifen zu können. SSH ist das gängigste Protokoll, das heutzutage für den Zugriff auf Geräte verwendet wird, da es eine sichere und verschlüsselte Verbindung mit dem Gerät bereitstellt. Einige der gängigen Terminalmonitore für den Zugriff auf die CLI der Router sind:

Festlegen von show-Befehlen oder Debug-Befehlen im Terminal Monitor

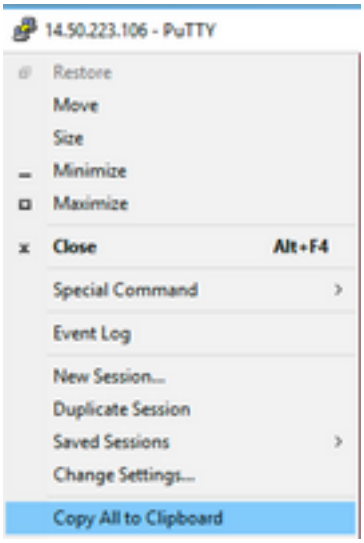
Es gibt verschiedene Möglichkeiten, die Ausgabe aus der CLI zu erfassen. Es wird empfohlen, die Informationen aus der CLI des Routers in eine separate Datei zu exportieren. Dies erleichtert die Weitergabe der Informationen an externe Parteien.

Es gibt folgende Möglichkeiten, die Ausgabe des Geräts zu erfassen:

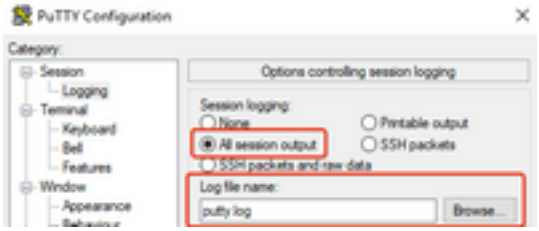
- Dump alle Ausgaben im Terminal, dafür müssen Sie sicherstellen, es gibt genügend Zeilen von scrollback, sonst der scrollback verpasst die ersten Abschnitte der Ausgabe und die Daten können unvollständig sein. Um die Scrollbacklinien in Putty zu erhöhen, navigieren Sie zu Putty Configuration > Window > Lines of Scrollback. Normalerweise ist dies auf einen sehr hohen Wert gesetzt, um genügend Scrollback-Ausgabe zu haben:



Später können Sie die Informationen vom Terminalmonitor mit der Option **Alle in Zwischenablage kopieren** erfassen und die Ausgabe in eine Textdatei einfügen:



- Eine weitere Option besteht darin, die gesamte Sitzungsausgabe in einer TXT-Datei zu protokollieren. Mit dieser Option werden alle eingegebenen Befehle und gesammelten Ausgaben sofort in der Textdatei protokolliert. Dies ist eine gängige Vorgehensweise, um alle Ausgaben in einer Sitzung zu protokollieren. Um alle Sitzungsausgaben in einer Datei unter Putty zu protokollieren, navigieren Sie zu **Putty Configuration > Session > Logging,** und wählen Sie dann All Session Output (Alle Sitzungsausgaben) wie folgt aus:



> **Anmerkung:** Wenn kein anderer Name angegeben ist, wird der Standardname für die Protokolldatei verwendet. Klicken Sie auf die Schaltfläche "Durchsuchen", um genau zu erfahren, wo die Datei gespeichert ist, damit Sie sie später finden können. Stellen Sie außerdem sicher, dass Sie keine andere Datei putty.log im gleichen Dateipfad überschreiben.

Sammeln der grundlegenden Ausgabe des Befehls show über die CLI

Show-Befehle sind erforderlich, um grundlegende Informationen vom Router zu sammeln, bevor eine Debugsammlung stattfindet. Show-Befehle sind schnell zu erfassen und haben größtenteils keine Auswirkungen auf die Leistung des Routers. Die Isolierung des Problems kann sofort mit der

Ausgabe des Befehls show beginnen.

Sobald der Router angeschlossen ist, kann die Anschlusslänge auf 0 gesetzt werden. Dies kann die Erfassung beschleunigen, um alle Ausgaben auf einmal anzuzeigen und die Verwendung der Leertaste zu vermeiden. Der einzige Befehl, der detaillierte Informationen über den Router sammelt, ist "show tech". Alternativ können Sie "**show tech voice**" erfassen, das Daten anzeigt, die spezifisch für die Sprachfunktionen sind, die auf dem Router aktiviert sind:

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

Erfassen der Debug-Ausgabe von der CLI

Die Erfassung der Debugausgabe in Cisco IOS/IOS-XE kann sich als problematisch erweisen, da die Gefahr eines Router-Absturzes besteht. Einige der Best Practices werden jedoch in den nächsten Abschnitten erläutert, um Probleme zu vermeiden.

Speicherüberprüfung

Bevor Sie Debug-Vorgänge aktivieren, müssen Sie sicherstellen, dass genügend Speicher vorhanden ist, um die Ausgabe im Puffer zu speichern.

Führen Sie den Befehl **show process memory** aus, um herauszufinden, wie viel Speicher Sie zuweisen können, um alle Ausgaben im Puffer zu protokollieren:

> **Tipp:** Verwenden Sie den Befehl **terminal length default** oder **terminal length <num_lines>,** um zu einer begrenzten Anzahl von im Terminal angezeigten Zeilen zurückzukehren.

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

Im Beispiel stehen 7666268552 Byte (7,6 GB) für die Verwendung durch den Router frei. Dieser Speicher wird vom Router von allen Systemprozessen gemeinsam genutzt, d. h. Sie können nicht den gesamten freien Speicher verwenden, um die Ausgabe im Puffer zu protokollieren, aber Sie können bei Bedarf eine gute Menge an Systemspeicher verwenden.

Die meisten Szenarien erfordern mindestens 10 MB, um genügend Debugausgabe zu sammeln, bevor die Ausgabe verloren geht oder überschrieben wird. In seltenen Fällen ist eine größere Datenmenge erforderlich, in diesen speziellen Szenarien können Sie 50MB bis 100MB im Puffer ausgeben oder Sie können höher gehen, solange der verfügbare Speicher zur Verfügung steht.

Wenn der freie Arbeitsspeicher nicht ausreichend ist, liegt möglicherweise ein Speicherleckproblem vor. Wenn dies der Fall ist, wenden Sie sich an das Architecture TAC-Team, um die Ursache für diesen geringen Arbeitsspeicher zu überprüfen.

Prüfung der zentralen Verarbeitungseinheit (CPU)

Die CPU wird durch die Anzahl der im System aktiven Prozesse, Funktionen und Anrufe beeinflusst. Je mehr Funktionen oder Anrufe im System aktiv sind, desto höher ist der Arbeitsaufwand für die CPU.

Ein guter Maßstab ist, sicherzustellen, dass der Router die CPU mit 30 % oder weniger hat, was bedeutet, dass Sie Debug-Vorgänge sicher von "Basic" bis "Advanced" aktivieren können (achten Sie immer auf die CPU, wenn "Advanced"-Debug verwendet wird). Liegt die CPU des Routers bei ca. 50 %, können grundlegende Fehlerbehebungen durchgeführt und die CPU sorgfältig überwacht werden. Wenn die CPU mehr als 80 % erreicht, stoppen Sie sofort die Fehlersuche (siehe weiter unten in diesem Artikel) und wenden Sie sich an das TAC.

Den **Prozess cpu sortiert** verwenden **| exclude 0.00** Befehl, um die letzten 5s, 60s und 5mins CPU Werte zusammen mit den oberen Prozessen zu überprüfen.

```
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
```

In der Ausgabe hat der Router nicht viel Aktivität, die CPU ist niedrig, und Debugging kann sicher aktiviert werden.

> **Vorsicht:** Achten Sie besonders auf die oberen aktiven CPU-Prozesse. Wenn die CPU 50 % oder höher ist und der obere Prozess ein Sprachprozess ist, können nur grundlegende Fehlerbehebungen aktiviert werden. Überwachen Sie die CPU kontinuierlich mit dem Befehl, um sicherzustellen, dass die Gesamtleistung des Routers nicht beeinträchtigt wird.

Prüfung der aktuellen aktiven Anrufe

Für jeden Router gelten andere Kapazitätsschwellenwerte. Es ist wichtig, die Anzahl der aktiven Anrufe im Router zu überprüfen, um sicherzustellen, dass die maximale Kapazität nicht erreicht wird. Das Datenblatt zu Cisco Unified Border Element Version 12 enthält Informationen zu den einzelnen Plattformkapazitäten.

Verwenden Sie den Befehl **show call active total-calls**, um eine Vorstellung davon zu erhalten, wie viele Anrufe im System aktiv sind:

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

Verwenden Sie den Befehl **show call active voice summary**, um detaillierte Informationen zu den einzelnen aktiven Anruftypen abzurufen:

```
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
```
Einige der gemeinsamen Werte sind:

- **Telefonische Gesprächsabschnitte**: TDM-Gateway-Anrufe, einschließlich analoger und PRI/ISDN-Anrufe
- **SIP-Anrufabschnitte**: SIP-Anrufe gesamt Wenn es sich um einen CUBE-Router handelt, werden zwei Anrufabschnitte pro Anruf angezeigt. Teilen Sie die hier angezeigten Anrufe durch 2, um eine genaue Zahl zu erhalten.
- **H323-Anrufabschnitte**: Gesamtzahl H323-Anrufe
- **SCCP-Anrufabschnitte**: Vom CUCM gesteuerte Medienressourcen, die im Router verwendet werden, z. B. Transcoder und MTPs.

Protokollierungspuffereinstellungen

Um den Router so zu konfigurieren, dass die Debug-Ausgabe im Puffer gespeichert wird, wird der Modus configure terminal aufgerufen, damit die Einstellungen in der CLI manuell angepasst werden können. Diese Konfiguration hat keine Auswirkungen auf den Router. Wie jedoch in den vorherigen Abschnitten beschrieben, ist der Befehl **show tech** oder **show running-config** des Routers erforderlich, falls ein Rollback der Konfiguration erforderlich ist.

Als Nächstes sehen Sie ein Konfigurationsbeispiel. Hierbei handelt es sich um eine gemeinsame Baseline, die von TAC-Technikern verwendet wird. In diesem Beispiel werden 10 MB Pufferspeicher zugewiesen. Dieser kann jedoch nach Bedarf erhöht werden:

```
# 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
```
Die Befehle führen folgende Aufgaben aus:

- **service timestamps debug oder log**: Stellt sicher, dass die lokale Router-Zeit in Millisekunden genau auf jede protokollierte Nachricht geschrieben wird. Dies ist nützlich, um Anrufe basierend auf der Zeit zu finden. Mit Millisekunden-Zeitstempeln können Sie Debug-Zeilen in logisch verwandte Ereignisse gruppieren, wenn zwei Zeilen innerhalb derselben Millisekunde auftreten.
- **Service-Sequenznummern:** Schreibt die Sequenznummer des Debugs in die Zeile. Dies ist nützlich (im Wesentlichen erforderlich), wenn Protokolle an einen Syslog-Server weitergeleitet werden. Dies ist sehr nützlich, um festzustellen, ob Debug-Meldungen an den Syslog-Server im Netzwerk verworfen wurden. Die Sequenznummer ist das erste Element im Debugging vor dem Zeitstempel und der eigentlichen Protokollmeldung. Beachten Sie, dass sich dies von dem Zeitstempel/der Sequenznummer unterscheidet, die Syslog-Server lokal in ihre Dateien schreiben können.
- **Protokollierungspuffer**: Weist den Router an, Debugging-Meldungen an seinen lokalen Pufferspeicher zu senden. Die Puffergröße wird in Byte festgelegt. In der Konfiguration wurde

die Puffergröße auf 10MB eingestellt.
- **Keine Protokollierungskonsole** und **kein Protokollierungsmonitor:** Auf der Konsole oder dem Terminalmonitor werden keine Protokollmeldungen ausgegeben. Wenn diese Befehle nicht konfiguriert werden, können sie sich negativ auf die Leistung des Routers und die Genauigkeit der Debugausgabe auswirken.
- **Voice Ice-Syslog:** Aktiviert Voice Internal Error Codes-Meldungen, um die Ursache für die Verbindungstrennung zu ermitteln.

Syslog-Einstellungen konfigurieren

Manchmal kann es sich um zufällige Probleme handeln, die eine Möglichkeit erfordern, Debugs kontinuierlich zu sammeln, bis das Ereignis eintritt. Wenn Sie die Debugs im Puffer speichern, werden sie kontinuierlich gesammelt. Beachten Sie, dass diese Option auf den Arbeitsspeicher beschränkt ist, den Sie zuweisen können. Sobald dieser Arbeitsspeicher erreicht ist, kreist der Puffer um den Arbeitsspeicher und verwirft die ältesten Nachrichten, was zu unvollständigen, wertvollen Informationen führt, die zur Isolierung des Problems erforderlich sind.

Mit Syslog kann der Router alle Debug-Meldungen an einen externen Server senden, wo die Syslog Server-Software sie in Textdateien speichert. Dies ist zwar eine gute Methode zum Erfassen der Debugausgabe, stellt jedoch nicht die bevorzugte Methode für die Protokollsammlung dar. Syslog-Server neigen dazu, aufgrund von Überlastung im Server Zeilen aus der empfangenen Ausgabe zu überspringen oder zu löschen, da die Debug-Ausgabe den Server überlasten kann oder Pakete aufgrund von Netzwerkbedingungen verworfen werden können. In einigen Szenarien ist Syslog jedoch die einzige Möglichkeit, Fortschritte bei einem Problem zu erzielen.

Verwenden Sie nach Möglichkeit eine zuverlässige Transportmethode wie TCP, um Datenverluste zu vermeiden, und schließen Sie den Syslog-Server als Vorschlag an denselben Switch an, an dem der Router angeschlossen ist, oder so nahe wie möglich am Router. Es garantiert immer noch nicht, dass alle Daten in den Dateien gespeichert werden, aber verringert die Wahrscheinlichkeit von Datenverlust.

Standardmäßig verwenden Syslog-Server UDP als Transportprotokoll auf Port 514.

```
#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
```

Sobald die Befehle konfiguriert sind, leitet der Router die Nachrichten sofort an die IP-Adresse des Syslog-Servers weiter.

Debug-Auflistung

Nachdem die Debugs aktiviert wurden, muss der Puffer gelöscht werden, bevor das Problem reproduziert wird. Auf diese Weise wird sichergestellt, dass die Ausgabe so sauber wie möglich ist und keine zusätzlichen Daten für die Analyse benötigt werden. Führen Sie den Befehl **clear log aus**, um sicherzustellen, dass der Puffer gelöscht wird. Wenn auf dem Router andere Anrufe aktiv sind und die Debug-Funktionen aktiviert sind, wird die Ausgabe sofort im Puffer ausgegeben.

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

Wenn das Problem reproduziert wurde, deaktivieren Sie das Debugging sofort, um weitere Ausgaben im Puffer zu stoppen. Sammeln Sie dann die Protokolle. Sie können alle Ausgaben im Terminal mit den folgenden Befehlen auslesen:

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

Manchmal wird PuTTY geschlossen, da es nicht alle Ausgaben auf einmal verarbeiten muss. Dies ist normal und bedeutet nicht, dass ein Fehler aufgetreten ist. Wenn dies geschieht, öffnen Sie die Sitzung erneut und fahren Sie normal fort. Wenn der Protokollierungspuffer zu groß ist oder der Terminalmonitor aufgrund der zu druckenden Datenmenge abstürzt, kopieren Sie die Pufferausgabe direkt mit dem Befehl **show log** auf ein externes Gerät. **| Weiterleiten**:

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

Der Befehl kopiert die gesamte Pufferausgabe in ein FTP mit der IP-Adresse 192.168.1.2 und dem Dateinamen debug.txt. Der Dateiname muss immer angegeben werden. Weitere für den Export dieser Daten verfügbare Ziele sind:

```
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
```

Welche Debugs können in Voice Routern aktiviert werden?

Jeder Anruffluss und jede Funktionsart (TDM, CUBE oder SCCP (Medienressourcen)) ist unterschiedlich, und es gibt spezifische Debugging-Optionen, die Sie aktivieren können. Alle erforderlichen Debugs müssen gleichzeitig aktiviert sein. Wenn jeweils nur ein Debugging erfasst wird, ist dies ineffektiv und sorgt bei der Analyse der Daten für mehr Verwirrung.

Die Debugging-Funktionen werden auf der CLI-exec-Eingabeaufforderungsebene **Router#** aktiviert. Hierfür müssen Sie über Berechtigungen für den privilegierten Ausführungsmodus

verfügen.

Es gibt einfache und erweiterte Debugging. Grundlegende Debugging-Funktionen werden zum Erfassen von Signalisierungsinformationen in SIP, H323 oder MGCP verwendet, um die Kommunikation zwischen dem Router und den Peer-Geräten darzustellen.

Erweiterte Debugs sind sehr detailliert und werden normalerweise verwendet, um im Fall von internen Stapelfehlern, die die grundlegenden Debugs nicht anzeigen können, mehr Informationen zu sammeln. Diese Fehlerbehebungen sind normalerweise CPU-intensiv.

> **Tipp:** Denken Sie daran, den Befehl **clear logging** auszuführen, nachdem die Debug-Funktionen aktiviert wurden. Mit diesem Befehl wird sichergestellt, dass der Puffer für eine sauberere Erfassung der Debugs gelöscht wird.

Debuggen der internen Anrufsteuerungs-API (CCAPI)

Innerhalb jedes Cisco IOS/IOS-XE Routers gibt es eine Anrufsteuerungs-API, die für die Kommunikation zwischen verschiedenen VoIP-Anwendungen oder -Protokollen und den Datenebenenkomponenten wie RTP, DSP, Voice Cards etc. zuständig ist. Um Daten von dieser Ebene zu erfassen, kann ein spezielles Debugging verwendet werden:

```
debug voip ccapi inout
```

Es gibt andere Optionen für dieses Debugging, jedoch deckt **debug voip ccapi inout** alle grundlegenden Wählplan- und Anrufeinrichtungsinformationen ab, die normalerweise mehr als genug sind, um die Zustände dieser Ebene zu verstehen.

> **Tipp: debug voip ccapi inout** hat in der Regel nur minimale Auswirkungen auf die CPU des Routers und wird empfohlen, zusammen mit allen Signalisierungs-Debugs aktiviert zu werden, um einen vollständigen Satz von Protokollen mit Informationen über den/die Anruf(e) und seine unterschiedlichen Zustände bereitzustellen.

SIP-Anrufflüsse

Diese Debug-Meldungen werden am häufigsten für SIP-Anrufverläufe verwendet und können innerhalb von CUBE- und TDM-Gateways mit einem SIP-Leg zwischen dem Router und dem CUCM oder einem anderen SIP-Server/Proxy aktiviert werden.

Grundlegende SIP-Fehlerbehebung

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

Erweiterte SIP-Fehlerbehebung

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

Digitale (PRI, BRI) Anrufflüsse

Diese Fehlerbehebungen gelten für Primary Rate Interfaces (PRI) T1/E1 oder Basic Rate Interfaces (BRI):

Grundlegendes digitales Debugging

```
debug isdn q931
```

Erweitertes digitales Debugging

```
debug isdn q921
```

Analoge Anrufflüsse

Diese Fehlerbehebungen werden verwendet, wenn es sich um analoge Leitungen wie FXS- (Foreign eXchange Subscriber) oder FXO-Ports (Foreign eXchange Office) handelt:

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

MGCP-Anrufflüsse

Diese Debug-Protokolle werden verwendet, wenn MGCP als Sprachprotokoll zwischen einem Sprach-Gateway und dem CUCM verwendet wird.

Grundlegende Debugs

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

CCM-Manager-Fehlerbehebung

Der **debugs ccm-manager** dient zum Nachverfolgen der Backhaul-Meldungen für Konfigurationsdownload, Warteschleifenmusik und PRI/BRI zwischen dem CUCM und dem Voice Gateway. Diese Debugs werden nach Bedarf verwendet und sind vom Fehlerszenario abhängig.

```
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
```

Erweiterte MGCP-Debugs

```
debug mgcp all
```

H323-Anrufflüsse

Obwohl H323 nicht häufig verwendet wird, gibt es dennoch einige Bereitstellungen mit

konfiguriertem H323:

Grundlegende H323-Debugs

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

Erweiterte H323-Debugs

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

SCCP-Medienressourcen

Diese Fehlerbehebungen dienen der Behebung von SCCP-Medienressourcenproblemen (Skinny Call Control Protocol), die MTP- (Media Termination Point) oder bei einem Cisco Unified Communications Manager- (CUCM-) Server registrierte Transcoder betreffen:

Grundlegende SCCP-Debugs

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

Erweitertes SCCP-Debugging

```
debug sccp all
```

VoIP-Verfolgung

Mit der Einführung von Cisco IOS-XE 17.4.1 und 17.3.2 gibt es eine neue Option zum Erfassen von Sprachprotokollen im Cisco Unified Border Element (CUBE). Diese neue Funktion heißt VoIP Trace. Hierbei handelt es sich um ein neues Framework für die Benutzerfreundlichkeit, mit dem SIP-Signalisierungen und -Ereignisse protokolliert werden, ohne dass Debugging-Vorgänge aktiviert werden müssen.

VoIP Trace ist standardmäßig aktiviert und kann bei Bedarf jederzeit deaktiviert werden. VoIP Trace erfasst nur bestimmte Informationen für SIP-Anrufe:

- SIP-Nachrichten für Anrufe vom SIP-Trunk zum Trunk
- Ereignisse und API-Anrufe vom SIP-Layer an andere Layer in CUBE
- SIP-Fehler
- Anrufsteuerung (Unified Communication-Anrufverläufe werden von CUBE verarbeitet)
- Status und Ereignisse von Finite State Machines (FSM)
- DFÜ-Peer zugeordnet
- Zugeordnete RTP-Ports
- IEC-Fehlerkorrelation mit SIP-Signalisierung

Einschränkungen

- VoIP Trace protokolliert keine Informationen zu Out-of-Dialog-SIP-Nachrichten: REGISTRIERENOPTIONENABONNIEREN/BENACHRICHTIGENINFORMATIONEN
- VoIP-Nachverfolgung in HA wird unterstützt, allerdings gelten folgende Einschränkungen: Auf dem Standby-Router ist VoIP Trace standardmäßig aktiviert. Nur anwendbare Ablaufverfolgungen für den Standby-Prozess werden angezeigt, bis er aktiv wird.Sobald der Standby-Modus aktiv ist, enthält er **KEINE** vollständigen Ablaufverfolgungen von zielgerichteten Anrufen und nur neue Anrufe.**show voip trace <key>** funktioniert weiterhin auf dem Standby-Router und zeigt Cover-Puffer- und Medien-Stream-Daten für Anrufe an

So aktivieren Sie VoIP Trace

Wie bereits erwähnt, ist diese Funktion standardmäßig aktiviert. Der Befehl zum Aktivieren dieser Funktion lautet:

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

Deaktivieren der VoIP-Ablaufverfolgung

Um diese Funktion zu deaktivieren, sind die folgenden Befehle verfügbar:

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

> **Vorsicht:** Nachdem die VoIP-Ablaufverfolgung deaktiviert wurde, wird der gesamte Speicher gelöscht, und Informationen gehen verloren.

Im Konfigurationsmodus für die Ablaufverfolgung stehen folgende Befehle zur Verfügung:

```
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
```

Speichergrenze konfigurieren

Die Speichergrenze bestimmt, wie viel Speicher von VoIP Trace zum Speichern der Daten verwendet wird. Standardmäßig sind 10 % des verfügbaren Speichers auf der Plattform verfügbar. Dies kann jedoch auf maximal 1 GB und mindestens 10 MB geändert werden. Der Speicher, der dynamisch zugewiesen wird, d. h. die Funktion verwendet nur den Speicher nach Bedarf und ist abhängig vom Anrufvolumen. Sobald der maximal verfügbare Speicher erreicht ist, kreist er um und löscht ältere Einträge.

Wenn der Speichergrenzwert so geändert wird, dass er größer als der verfügbare Speicher von 10

% ist, wird in der Befehlszeilenschnittstelle eine Meldung angezeigt:

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

Um die Standardspeicherauslastung von 10 % festzulegen, kann der Befehl **memory-limit platform** verwendet werden:

```
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]:
```

> **Vorsicht:** Wenn die Speichergrenze reduziert wird, gehen alle VoIP-Trace-Daten verloren. Bevor der Speicher verkleinert wird, muss eine Sicherung der Daten erfolgen.

Anzeigen von VoIP-Ablaufverfolgungsdaten

Um die Daten von VoIP Trace anzuzeigen, müssen spezifische Befehle zum Anzeigen verwendet werden. Die Daten können in derselben Terminalsitzung angezeigt oder auch über Syslog an einen externen Syslog-Server gesendet werden.

> **Anmerkung:** Ablaufverfolgungen werden nach 32 Sekunden nach dem Empfang eines BYE für einen Anruf gelöscht.

> **Anmerkung:** Die SIP-Signalisierung wird pro Leg angezeigt und nicht wie bei regulären Debugs kombiniert. Regelmäßige Debug-Vorgänge wie **debug csip-Meldungen** zeigen die SIP-Signalisierung eines Anrufs in der genauen Reihenfolge der Ereignisse an. In VoIP Trace ist jeder Abschnitt separat. Zur Bestimmung der richtigen Reihenfolge werden die Zeitstempel verwendet.

Folgende Befehle stehen zur Anzeige der Daten zur Verfügung:

```
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
```

voip trace all anzeigen

Dieser Befehl zeigt alle im Puffer verfügbaren VoIP-Trace-Daten an. Die Verwendung dieses Befehls beeinträchtigt die Leistung des Routers. Nach Eingabe des Befehls wird eine Warnmeldung angezeigt, die Sie über das Risiko informiert und den Fortgang bestätigt:

```
Router# show voip trace all
 Displaying 11858 cover buffers
```

```
 This may severely impact system performance.
 Continue? [yes/no] no
```

show voip trace cover-buffers

Dieser Befehl zeigt eine Übersicht der Anrufdetails für alle unter VoIP-Trace gemeldeten Anrufe an. Für jeden Anrufabschnitt wird ein Cover-Puffer erstellt, der eine Zusammenfassung des protokollierten Anrufs enthält.

```
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
----------------------------------------------
```

Weitere Informationen zu den einzelnen Feldern finden Sie in der folgenden Tabelle:

Feld	Beschreibung
Suchschlüssel	Enthält eine Kombination aus Anrufer, angerufener Nummer und Anruf-ID
Zeitstempel	Erstellungszeit des Abdeckungspuffers
Puffer-ID	Pufferkennung des Deckungspuffers
Anruf-ID	Call-ID des jeweiligen Call-Abschnitts von zum Cover-Puffer
Peer-Anruf-ID	Anruf-ID des Peer-Abschnitts
Korrelator	FPI-Korrelator des Anrufs
Angerufene Nummer	Angerufene Nummer des jeweiligen Anrufabschnitts des Abdeckungspuffers
Anrufernummer	Rufnummer der jeweiligen Rufstrecke des Deckungspuffers
SIP-Anruf-ID	SIP-Anruf-ID des jeweiligen Anrufabschnitts des Abdeckungspuffers
SIP-Sitzungs-ID	SIP-Sitzungs-ID des jeweiligen Anrufabschnitts des Abdeckungspuffers
GUID	GUID des jeweiligen Rufs des Deckungspuffers
Ankerbein	Der Ankerzweig wird auf "yes" (Ja) gesetzt, wenn der jeweilige Anrufzweig ein Ankerzweig im Anrufweiterleitungsprozess oder in der M Proxy-Bereitstellung ist.
Gabelbein	Forked Leg wird auf yes gesetzt, wenn die jeweilige Anrufstrecke eine Ankerstrecke im Anrufweiterleitungsprozess oder in der Media P Bereitstellung ist.
Zugehörige Anruf-IDs	Call-ID der zugehörigen Gabelbeine

Um die Cover-Puffer zu filtern, können Sie die Befehle **include** und **section** verwenden:

```
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

In Kombination mit dem vorherigen Befehl kann **show voip trace call-id** verwendet werden, um die Aufrufe zu finden. Nachdem die Anruf-ID identifiziert wurde, kann dieser Befehl verwendet werden, um alle Informationen zum jeweiligen Anrufabschnitt anzuzeigen:

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

Zeigt VoIP-Ablaufverfolgungsstatistiken an

Dieser Befehl show zeigt detaillierte Informationen zu Status, Speicherbelegung, fehlerhaften oder fehlerhaften Aufrufen, erfolgreichen Aufrufen, Zeitstempeln der neuesten und ältesten Einträge und mehr an.

```
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
```

Weitere Informationen zu den einzelnen Feldern finden Sie in der folgenden Tabelle:

Feld	Beschreibung
Ablaufverfolgungsstatus	Zeigt den Ablaufverfolgungsstatus an, einschließlich Datum und Uhrzeit, zu der die VoIP-Ablaufverfolgung aktiviert wurde.
Speicherlimit konfiguriert	Zeigt das konfigurierte Speicherlimit an. Dies entspricht 10 % der Speichergröße des Prozessorpools.
Arbeitsspeicher belegt	Zeigt den dynamisch für VoIP-Trace belegten Speicher an
Gesamtzahl abgebrochener Anrufverbindungen	Zeigt die Anzahl der fehlerhaften Anrufabschnitte an, die in den Protokollierungspuffer geschrieben wurden. Gedumpte Anrufe beziehen Anrufabschnitte, die mit IEC-Fehlern verbunden sind
Älteste Spurensicherung	Zeigt Zeitstempel und Suchschlüssel für den ältesten fehlgeschlagenen Anruf seit Aktivierung der VoIP-Ablaufverfolgung an.
Neueste Spur verworfen	Zeigt Zeitstempel und Suchschlüssel für den letzten fehlgeschlagenen Anruf seit Aktivierung der VoIP-Ablaufverfolgung an.
Gesamtzahl erfasster Anrufabschnitte	Zeigt die Gesamtzahl der nach Aktivierung von VoIP Trace erfassten Levels an
Verfügbare Anrufabschnitte gesamt	Zeigt die Gesamtzahl der verfügbaren Anrufabschnitte im Verlauf an. Dies kann im Vergleich zur Gesamtanzahl der erfassten Anrufabschnitte gleich oder unterschiedlic sein, je nach Speicherlimit.
Älteste verfügbare Spur	Zeigt den Zeitstempel und den Suchschlüssel des ältesten verfügbaren Deckungspu im Speicher an.

Neueste Spur verfügbar	Zeigt den Zeitstempel und den Suchschlüssel für den neuesten im Speicher verfügba Cover-Puffer an.
Verpasste Ablaufverfolgungen gesamt	Zeigt die Anzahl der Anrufabschnitte an, die aufgrund eines Speicherlimits verpasst wurden.

Zusätzliche show-Befehle

Feld	Nutzung	
show voip trace correlator <Korrelator>	show voip trace correlator 4	Filtert und zeigt VoIP-Ablaufverfolgung für eine bestimmte An
show voip trace session-id <Sitzungs-ID>	show voip trace session-id 87003120822b5dbd8fd80f62d8e57c48	Filtert und zeigt VoIP-Ablaufverfolgung lokale oder die Remote-UUID aus den werden, um beide Abschnitte des Anru
show voip trace sip-call-id <Anruf-ID>	show voip trace sip-call-id 01e60dfa9d8442848336d79e3155a8a1	Filtert und zeigt VoIP-Ablaufverfolgung

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.