Configureer debug Collection voor Unified border element (CUBE) en Time-Division Multiplexing (TDM) gateways

Inhoud

Inleiding

Dit document beschrijft een aantal van de best practices om spraakdebugs in een Cisco IOS/IOS-XE spraakrouter te verzamelen.

Voorwaarden

De informatie in dit document is gebaseerd op de apparaten in een specifieke laboratoriumomgeving. Alle apparaten die in dit document worden beschreven, hadden een opgeschoonde (standaard)configuratie. Als uw netwerk live is, moet u zorgen dat u de potentiële impact van elke opdracht begrijpt.

Vereisten

- Basiskennis in Cisco IOS/IOS-XE binnen geïntegreerde services routers (ISR).
- Gepersonaliseerde toegang om opdrachten in de ISR-routers uit te voeren.
- Eerdere ervaring met Voice-over-IP (VoIP)-protocollen is gewenst.
- Voor VoIP Trace is minimaal Cisco IOS-XE 17.4.1 of 17.3.2 vereist.

Gebruikte componenten

Voor de toepassing van dit document worden de volgende componenten gebruikt:

- Cisco ISR 3925 router
- Cisco ISR 4451 router
- PuTTY

De informatie in dit document is gebaseerd op de apparaten in een specifieke laboratoriumomgeving. Alle apparaten die in dit document worden beschreven, hadden een opgeschoonde (standaard)configuratie. Als uw netwerk live is, moet u zorgen dat u de potentiële impact van elke opdracht begrijpt.

Achtergrond

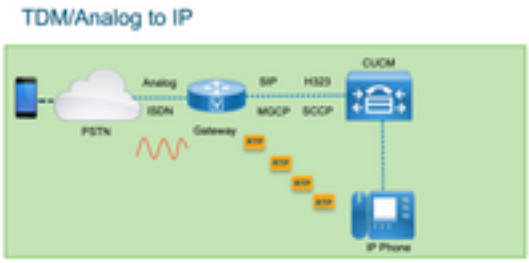
Het proces van Debug collectie in deze platforms heeft uitdagingen en zou de prestaties van het apparaat kunnen beïnvloeden. De uitdagingen en risico's nemen toe wanneer er meerdere actieve

oproepen zijn die in een spraakrouter zijn gemaakt. In sommige scenario's als de debugs niet correct worden verzameld, kan het tot hoge CPU leiden die de capaciteit van de router zou kunnen schaden en zelfs een softwarerecrisis veroorzaken. Dit document heeft betrekking op het verschil tussen een Cisco Unified Border Element (CUBE) en een TDM/Analog Gateway.

TDM-spraakgateways versus CUBE

TDM-spraakgateways worden voornamelijk gebruikt om een intern telefoonsysteem te verbinden met een andere Private Branch Exchange (PBX) of het Public Switched Telephony Network (PSTN). Het type verbindingen dat in TDM-gateways wordt gebruikt, is T1/E1-controllers (ISDN of CAS) en analoge circuits zoals FXS- en FXO-poorten. Een Digital Signal Processor (DSP) converteert het geluid van zijn onbewerkte vorm naar RTP-pakketten. Op een gelijkaardige manier, worden de pakketten van RTP omgezet in ruwe audio nadat DSP de pakketten van RTP heeft verwerkt en de audio op de specifieke kring verzendt. Deze gateways kunnen interworking met H323, MGCP of SCCP aan de VoIP-kant, en aan de TDM-kant zijn ISDN PRI-circuits of analoog als de meest gebruikelijke verbindingen met het PSTN of de eindpunten.

Zoals in het beeld wordt getoond, bieden de TDM-gateways een brug tussen uw interne VoIP-infrastructuur en de analoge of ISDN-serviceproviders.



Met de introductie van VoIP begonnen klanten hun bestaande systemen snel te veranderen in een moderne VoIP-infrastructuur. Hetzelfde gebeurde aan de kant van de serviceproviders, waar zij nu verbindingen gebruiken om on-Premises telefonieservices te verbinden met de VoIP-infrastructuur voor serviceproviders en hun mogelijkheden uit te breiden om betere services te kunnen leveren. Het meest gebruikte VoIP-protocol is het Session Initiation Protocol (SIP) en wordt momenteel wereldwijd gebruikt door klanten en Internet Telephony Service Providers (ITSP).

CUBE is geïntroduceerd om een manier te bieden om die interne VoIP-systemen met de externe wereld te verbinden via de ITSP's met SIP als het primaire VoIP-protocol. CUBE is gewoon een IP-IP gateway waarvoor het niet langer een TDM-type verbinding nodig heeft, zoals T1/E1-controllers of analoge poorten. CUBE wordt uitgevoerd op dezelfde platforms als TDM Gateways.

Het meest gebruikte VoIP-protocol is SIP, voor het instellen en verwijderen van oproepen, en RTP voor mediatransport. In CUBE is geen DSP nodig tenzij een transcoder vereist is. De RTP-verkeersstromen eindigen van de ITSP naar het eindpunt, en CUBE fungeert als de tussenpersoon met adresverbergen als een van de vele functies die het biedt.

Zoals in het beeld wordt getoond, biedt CUBE een scheiding tussen uw interne VoIP-infrastructuur en de SIP ITSP:

Verzameling van Cisco IOS/IOS-XE spraakdebugs

Spraakfuncties die worden uitgevoerd op een andere lijst met platforms, zoals ISR, ASR's, CAT8Ks en andere, maar waarbij een algemene software wordt gebruikt die Cisco IOS of Cisco IOS-XE is (de verschillen tussen Cisco IOS en Cisco IOS-XE worden niet in dit artikel behandeld). Laten we beginnen met de basisbeginselen voor toegang tot de Cisco IOS-router.

Toegang tot een Cisco IOS/IOS-XE router via Command Line Interface (CLI)

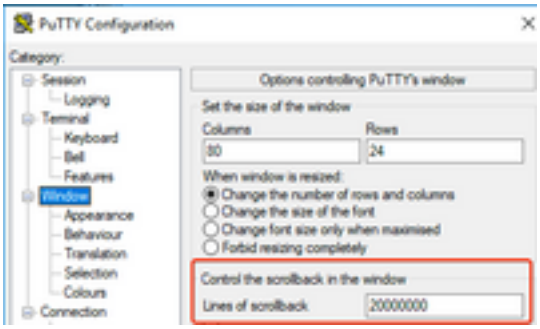
Routers hebben, net als alle andere op CLI gebaseerde apparaten, een eindmonitor nodig om toegang te krijgen tot de opdrachten via Secure Shell (SSH) of Telnet. SSH is het meest gebruikelijke protocol dat tegenwoordig wordt gebruikt om toegang te krijgen tot de apparaten, omdat het een beveiligde en versleutelde verbinding met het apparaat biedt. Enkele gemeenschappelijke eindmonitoren die worden gebruikt om tot CLI van de Routers toegang te hebben zijn:

Hoe de te verzamelen Terminalmonitor in te stellen, toont opdrachten of debugs

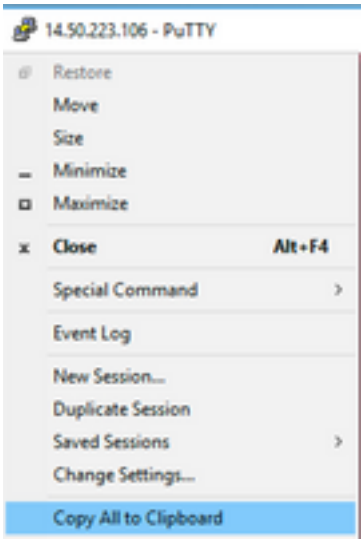
Er zijn verschillende manieren om de uitvoer van de CLI te verzamelen. Het is aan te raden om de informatie van de CLI van de router te exporteren naar een afzonderlijk bestand. Dit maakt het makkelijker om de informatie te delen met derden.

Een paar manieren om de uitgangen van het apparaat te verzamelen zijn:

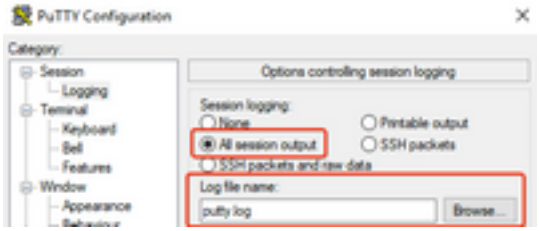
- Dump alle output in de terminal, voor dit moet u ervoor zorgen dat er genoeg regels van scrollback zijn, anders mist scrollback de eerste secties van de output en de gegevens kunnen onvolledig zijn. Om de scrollback lijnen in Putty te verhogen, navigeer naar Putty Configuration > Window > Lijnen van Scrollback. Normaal wordt dit ingesteld op een zeer hoge waarde om voldoende scrollback-uitvoer te hebben:



Later kunt u de informatie van de eindmonitor met het **Exemplaar allen aan de** optie van het **Klembord** verzamelen en de output kleven in een tekstdossier:



- Een andere optie is om de gehele sessie uitvoer te registreren naar een .txt bestand. Met deze optie worden alle opdrachten die worden ingevoerd en alle resultaten die worden verzameld, onmiddellijk aangemeld bij het tekstbestand. Dit is een veel voorkomende praktijk om alle uitvoer in een sessie te registreren. Ga als volgt te werk om alle sessieuitvoer naar een bestand in Putty te registreren **door** naar **Putty Configuration > Session > Logging** en selecteer vervolgens Alle sessieuitvoer:



> **Opmerking:** De standaard Logbestandsnaam wordt gebruikt als er geen andere naam is opgegeven. Klik op de knop Bladeren om precies te weten waar het bestand wordt opgeslagen om het later te vinden. Zorg er ook voor dat u niet een ander putty.log bestand overschrijven in hetzelfde bestand pad.

Verzamel basis tonen beveloutput van CLI

Toon bevelen zijn nodig om basisinformatie van de router te verzamelen alvorens om het even welke debug inzameling plaatsvindt. Toon opdrachten snel te verzamelen zijn, en voor het grootste deel, hebben geen invloed in prestaties op de router. Isolatie van het probleem kon onmiddellijk met enkel een output van het showbevel beginnen.

Zodra verbonden met de router, kan de eindlengte aan 0 worden geplaatst. Dit kan de inzameling sneller maken om alle output meteen te tonen, en het gebruik van de ruimtebar te vermijden. De ene opdracht die gedetailleerde informatie over de router verzamelt is 'show tech', en u kunt ook **tonen tech spraak** die gegevens specifieker toont voor de spraakfuncties die in de router zijn ingeschakeld:

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

Verzamel debug-uitvoer van de CLI

Debug uitvoerverzameling in Cisco IOS/IOS-XE kan soms een uitdaging zijn omdat er een risico op een routercrash is. Enkele van de best practices worden in de volgende secties uitgelegd om problemen te voorkomen.

Geheugencontrole

Alvorens u om het even welke debugs toelaat, moet u ervoor zorgen er genoeg geheugen is om de output in de buffer op te slaan.

Voer de opdracht **tonen procesgeheugen** om te weten te komen hoeveel geheugen u kunt toewijzen om alle uitvoer in de buffer te registreren:

> **Tip**: Gebruik de opdrachtterminallengte **standaard** of **eindlengte <num_lines>** om terug te gaan naar een beperkt aantal lijnen weergegeven in de terminal.

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

In het voorbeeld, is er 7666268552 bytes (7.6GB) vrij om door de router worden gebruikt. Dit geheugen wordt gedeeld door de router onder alle systeemprocessen, dit betekent dat u niet het gehele vrije geheugen kunt gebruiken om de output in de buffer te registreren, maar u kunt een goede hoeveelheid systeemgeheugen gebruiken zoals nodig.

De meeste scenario's vereisen minstens 10MB om genoeg te verzamelen debug uitvoer alvorens de output wordt verloren of overschreven. In zeldzame gevallen is een grotere hoeveelheid gegevens nodig om te worden verzameld, in die specifieke scenario's kunt u 50MB tot 100MB aan output in de buffer krijgen of u kunt hoger gaan zolang er geheugen beschikbaar is.

Als het Vrije Geheugen laag is, dan is er potentieel een probleem van het geheugenlek, als dit het geval is, gelieve het team van de Architectuur TAC in te schakelen om te herzien wat de oorzaak van dergelijk laag geheugen zou kunnen zijn.

Central Processing Unit (CPU)-controle

De CPU wordt beïnvloed door de hoeveelheid processen, functies en aanroepen die in het systeem actief zijn. Hoe meer functies of aanroepen actief zijn in het systeem, hoe drukker de

CPU is.

Een goede benchmark is te verzekeren dat de router de CPU bij 30% of minder heeft, wat betekent dat u veilig debugs van basis naar geavanceerd kunt inschakelen (houd altijd een oogje op de CPU wanneer geavanceerde debugs worden gebruikt). Als de router CPU ongeveer 50% bedraagt, kunnen eenvoudige debugs worden uitgevoerd en kan de CPU zorgvuldig worden bewaakt. Als de CPU hoger is dan 80%, stopt u de debugs (zie verderop in dit artikel) en neemt u TAC in voor ondersteuning.

Gebruik het **cpu** van het **showproces gesorteerd | sluit 0.00** uit om de laatste 5s, 60s en 5min CPU-waarden te controleren, samen met de bovenste Processen.

```
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 de output, heeft de router niet veel activiteit, is cpu laag, en debugs kan veilig worden toegelaten.

> **Voorzichtig:** Besteed extra aandacht aan de belangrijkste actieve CPU-processen, als de CPU 50% of hoger is en het hoogste proces een spraakproces is, kunnen alleen eenvoudige debugs worden ingeschakeld. Controleer de CPU voortdurend met de opdracht om er zeker van te zijn dat de algemene prestaties van de router niet worden beïnvloed.

Huidige controle van actieve oproepen

Elke router heeft verschillende capaciteitsdrempels. Het is belangrijk om te controleren hoeveel oproepen actief zijn in de router om te verzekeren dat het niet dicht bij maximale capaciteit is. Het gegevensblad Cisco Unified Border Element versie 12 biedt informatie over elke platformcapaciteit voor raadpleging.

Gebruik het bevel van de **showvraag actieve totaal-vraag** om een idee te krijgen op hoeveel vraag in het systeem actief is:

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

Gebruik het bevel van de **showvraag actieve stemsamenvatting** om een gedetailleerdere informatie van de specifieke vraagtypes te krijgen die actief zijn:

```
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
```

Enkele gemeenschappelijke waarden zijn:

- **Bel-benen voor telefonie**: TDM-gatewaygesprekken, waaronder analoge en PRI/ISDN-gesprekken.
- **SIP-aanroepbenen**: Totale SIP-oproepen. Als dit een CUBE router is, dan toont dit 2 vraagbenen per vraag. Verdeel de hier getoonde oproepen door 2 om een accuraat nummer te krijgen.
- **H323 aanroepbenen**: Totale H323-oproepen.
- **SCCP aanroepbenen**: CUCM Controlled Media Resources die in de router worden gebruikt, zoals Transcoder en MTP's.

Instellingen logbuffer

Om de router te vormen om op te slaan debug uitvoer in de buffer, wordt de configuratie terminal mode ingevoerd om de instellingen in de CLI handmatig te knijpen. Deze configuratie heeft geen invloed op de router, maar zoals in vorige secties wordt getoond, **toon technologie** of **toon in werking stelt -in werking stellen-config** bevel van de router is nodig in het geval dat de configuratie moet worden gerold.

Een configuratievoorbeeld kan hierna worden weergegeven, wat een veel gebruikte basislijn is voor TAC Engineers. Het voorbeeld wijst een 10MB buffergeheugen toe maar het kan worden verhoogd zoals nodig:

```
# 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
```

De opdrachten vervullen deze taken:

- **debug of log voor servicetijdstempels**: Zorgt ervoor dat de lokale routertijd op elk geregistreerd bericht, met milliseconde nauwkeurigheid wordt geschreven. Dit is nuttig om vraag te vinden die op tijd wordt gebaseerd. Met milliseconde tijdstempels kunt u debug-regels in logische gerelateerde gebeurtenissen groeperen wanneer twee regels binnen dezelfde milliseconde voorkomen.
- **volgnummers van de dienst:** Schrijft het volgnummer van de debug in de regel. Dit is nuttig (hoofdzakelijk vereist) wanneer de logboeken aan een syslogserver door:sturen. Dit zeer nuttig om te identificeren als om het even welke debug berichten aan de syslogserver in het netwerk zijn gelaten vallen. Het volgnummer is het eerste item in de debug, voor de tijdstempel en het eigenlijke logbericht. Merk op dat dit verschillend is van de timestamp/opeenvolgingsaantal syslog servers kunnen plaatselijk in hun dossiers schrijven.
- **logboekbuffer**: Vertelt de router om debugs naar zijn lokale buffergeheugen te verzenden. De buffergrootte wordt ingesteld in bytes. In de configuratie werd de buffergrootte ingesteld op 10MB.
- **geen logboekconsole** en **geen logboekmonitor:** Er worden geen logberichten afgedrukt in de console of terminal monitor. Als deze opdrachten niet zijn geconfigureerd, kunnen ze schadelijk zijn voor de routerprestaties en debug uitvoernauwkeurigheid.

- **voice iec syslog:** Laat de berichten van de Foutcodes van de Stem toe om te bepalen redenen losmaakt.

Syslog-instellingen configureren

Soms kunnen problemen willekeurig zijn en vereisen een manier om continu debugs te verzamelen tot de gebeurtenis gebeurt. Wanneer u de debugs in de buffer opslaat, verzamelt het deze continu. Merk op dat het beperkt is tot de hoeveelheid geheugen die u kunt toewijzen en zodra het die hoeveelheid geheugen bereikt, de buffer cirkelt rond en laat vallen de oudste berichten, die tot onvolledige waardevolle informatie leidt die nodig is om het probleem te isoleren.

Met Syslog kan de router alle debug berichten naar een externe server sturen, waar de Syslog Server software het in tekstbestanden opslaat. Hoewel het een goede manier is om de debug-uitvoer te verzamelen, is het niet de voorkeursmethode voor het verzamelen van logbestanden. Syslog Servers hebben de neiging om lijnen van de ontvangen uitvoer over te slaan of te laten vallen vanwege congestie in de Server, omdat debug-uitvoer de server kan overweldigen, of pakketten kunnen laten vallen vanwege netwerkomstandigheden. In sommige scenario's is Syslog echter de enige manier om vooruitgang te boeken in een kwestie.

Als het mogelijk is, gebruik dan een betrouwbare transportmethode zoals TCP om verlies van informatie te voorkomen en als suggestie de Syslog-server te verbinden met dezelfde switch waar de router is aangesloten of zo dicht mogelijk bij de router. Het garandeert nog steeds niet dat alle gegevens in de bestanden worden opgeslagen, maar vermindert de kans op gegevensverlies.

Standaard gebruiken syslog servers UDP als transportprotocol op poort 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
```

Zodra de opdrachten zijn geconfigureerd, stuurt de router de berichten onmiddellijk door naar het IP-adres van de Syslog-server.

Debug Collection

Zodra de debugs zijn ingeschakeld, moet de buffer worden gewist voordat het probleem wordt gereproduceerd. Dit wordt gedaan om ervoor te zorgen dat de output zo schoon mogelijk is en om te voorkomen dat extra gegevens nodig zijn voor de analyse. Laat het commando **clear log**

draaien, dit zorgt ervoor dat de buffer wordt gewist. Als er andere aanroepen actief zijn in de router en de debugs zijn ingeschakeld, wordt de output onmiddellijk in de buffer afgedrukt.

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

Nadat het probleem is gereproduceerd, schakelt u de debugs direct uit om meer output in de buffer te stoppen. Dan verzamel de logboeken. U kunt alle uitvoer in de terminal dumpen met de opdrachten:

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

Soms sluit PuTTY aangezien het niet alle output meteen behandelt, is dit normaal en het betekent geen mislukking is gebeurd, als dit gebeurt heropen de zitting opnieuw en ga normaal verder. In scenario's waar de logboekbuffer te groot is of de eindmonitor crasht vanwege de hoeveelheid gegevens die moet worden afgedrukt, kopieer de bufferoutput naar een extern apparaat direct met het bevel **tonen logbestand | doorsturen**:

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

De opdracht kopieert de gehele bufferoutput naar een ftp met IP-adres 192.168.1.2 met de bestandsnaam debug.txt. Bestandsnaam moet altijd worden opgegeven. Andere voor uitvoer beschikbare bestemmingen zijn:

```
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
```

Welke Debugs kan in Voice Routers worden ingeschakeld?

Elke aanroepstroom en elk type functies (TDM, CUBE of SCCP (Media Resources)) zijn verschillend en er zijn specifieke debugs die u kunt inschakelen. Alle vereiste debugs moeten tegelijkertijd worden ingeschakeld. Wanneer slechts één debug tegelijkertijd wordt opgenomen, is deze ineffectief en geeft dit meer verwarring wanneer de gegevens worden geanalyseerd.

Debugs zijn ingeschakeld binnen de CLI exec prompt level **Router#** die vereist dat u geprivilegieerde machtigingen voor de uitvoeringsmodus hebt.

Er zijn basis en geavanceerde debugs. Basis debugs worden gebruikt om signaleringsinformatie te verzamelen in SIP, H323 of MGCP, dat de gesprekken toont die de router heeft met zijn peer-apparaten.

Geavanceerde debugs zijn zeer gedetailleerd en worden normaal gebruikt om meer informatie te verzamelen in het geval van interne stack fouten die de basis debugs niet kunnen tonen. Deze debugs zijn gewoonlijk CPU-intensief.

> **Tip**: Nadat de debugs zijn ingeschakeld, vergeet niet om de opdracht **clear logging** uit te voeren. Deze opdracht zorgt ervoor dat de buffer wordt gewist voor een schonere opname van de debugs.

Debug voor interne Call Control API (CAPI)

Binnen elke Cisco IOS/IOS-XE router is er een Call Control API die verantwoordelijk is voor de communicatie tussen verschillende VoIP-toepassingen of -protocollen en de componenten van het gegevensplane, zoals RTP, DSP, spraakkaarten enzovoort. Om gegevens van deze laag op te nemen is er één specifieke debug die kan worden gebruikt:

```
debug voip ccapi inout
```

Er zijn andere opties voor dit debug, maar **debug voip capi inout** dekt alle basis kiesschema en vraag oprichting informatie die normaal meer dan genoeg is om te begrijpen wat de staten van deze laag zijn.

> **Tip**: **debug voip capi inout** heeft meestal een minimale impact op de CPU van de router en wordt aanbevolen om ingeschakeld te worden samen met signalering debugs om een volledige set logbestanden te voorzien van informatie over de oproep(en) en de verschillende toestanden.

SIP-gespreksstromen

Deze debugs zijn de meest gebruikte voor SIP-gespreksstromen en kunnen worden ingeschakeld binnen CUBE- en TDM-gateways met een SIP-been tussen de router en CUCM of een andere SIP-server/proxy.

Basis SIP-debugs

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

Geavanceerd SIP-debugs

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

Digitale gespreksstromen (PRI, BRI)

Deze debugs zijn van toepassing op Primary Rate Interfances (PRI) T1/E1 of Basic Rate Interfaces (BRI):

Basis digitale debug

```
debug isdn q931
```

Geavanceerde digitale debug

```
debug isdn q921
```

Analoge gespreksstromen

Deze debugs worden gebruikt wanneer er analoge circuits betrokken zijn zoals FXS- (Foreigh eXchange Subscriber) of FXO-poorten (Foreign eXchange Office):

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

MGCP-gespreksstromen

Deze debugs worden gebruikt wanneer MGCP gebruikt wordt als het Voice Protocol tussen een spraakgateway en CUCM.

Basis debugs

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

Debugs van CCM-Manager

De **debugs ccm-manager** wordt gebruikt om de configuratie download, MoH en PRI/BRI backhaul berichten tussen CUCM en de spraakgateway te volgen. Deze debugs worden gebruikt op de gewenste basis en zijn afhankelijk van het falen scenario.

```
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
```

Geavanceerde MGCP-debugs

```
debug mgcp all
```

H323 gespreksstromen

Hoewel H323 niet veel wordt gebruikt, zijn er nog steeds enkele implementaties met H323 geconfigureerd:

Basis H323 debugs

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

Geavanceerde H323-debugs

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

SCP-mediabronnen

Deze debugs worden gebruikt voor het oplossen van problemen met Skinny Call Control Protocol (SCCP) Media Resources die betrekking hebben op Media Termination Point (MTP) of Transcoders die zijn geregistreerd op een Cisco Unified Communications Manager (CUCM) server:

Basis SCP-debugs

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

Geavanceerd SCP-debug

```
debug sccp all
```

VoIP-tracering

Met de introductie van Cisco IOS-XE 17.4.1 en 17.3.2 is er een nieuwe optie om spraaklogs binnen het Cisco Unified Border Element (CUBE) op te nemen. Deze nieuwe functie wordt VoIP Trace genoemd. Dit is een nieuw servicability framework gemaakt om SIP signalering en gebeurtenissen te loggen zonder de noodzaak om debugs in te schakelen.

VoIP Trace is standaard ingeschakeld en kan naar behoefte op elk moment worden uitgeschakeld. VoIP Trace neemt alleen specifieke informatie op voor SIP-oproepen:

- SIP-berichten voor SIP-trunk naar Trunk-oproepen
- Gebeurtenissen en API-aanroepen van SIP-laag naar andere lagen in CUBE
- SIP-fouten
- Gespreksbeheer (Unified Communications CallFlow), verwerkt door CUBE
- Finite State Machines (FSM) staten en gebeurtenissen
- Gekoppelde dial-peers
- RTP-poorten toegewezen
- IEC-fouten in correlatie met SIP-signalering

Beperkingen

- VoIP Trace logt geen informatie in met betrekking tot uit-van-dialoog SIP-berichten: REGISTRERENOPTIESABONNEREN/MELDENINFORMATIE
- VoIP Trace in HA wordt ondersteund, maar deze voorbehouden zijn van toepassing: Standby router heeft VoIP Trace standaard ingeschakeld. Alleen toepasbare sporen voor het Standby-proces worden weergegeven totdat het actief wordtZodra de Standby-functie actief is, bevat deze **GEEN** volledige sporen van checkpointoproepen en alleen nieuwe oproepen**tonen voip-**

spoor <key> werkt nog steeds aan de standby-router en geeft dekking buffer en media stream gegevens voor oproepen weer

VoIP-tracering inschakelen

Zoals vermeld, wordt deze functie standaard ingeschakeld. De opdracht om deze functie in te schakelen is:

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

VoIP-tracering uitschakelen

Om deze functie uit te schakelen, zijn de opdrachten:

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

Voorzichtig: Nadat VoIP Trace is uitgeschakeld, wordt al het geheugen gewist en wordt informatie verloren.

De opdrachten die beschikbaar zijn in de modus voor overtrekken zijn:

```
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
```

Geheugenlimiet instellen

De geheugenlimiet bepaalt hoeveel geheugen door VoIP Trace wordt gebruikt om de gegevens op te slaan. Standaard is 10% van het beschikbare geheugen in het platform, maar dit kan worden veranderd in een max van 1 GB en een min van 10MB. Het geheugen wordt dynamisch toegewezen, wat betekent dat de functie alleen geheugen gebruikt als dat nodig is en afhankelijk is van het volume van de oproep. Zodra het de max beschikbare geheugen bereikt, cirkelt het rond en verwijdert oudere items.

Wanneer de geheugenlimiet is gewijzigd zodat deze groter is dan het 10% beschikbare geheugen, wordt een bericht weergegeven in de Command Line Interface:

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

Om de standaardinstelling van 10% geheugengebruik in te stellen, kan het commando **memory-limit platform** worden gebruikt:

```
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]:
```

Voorzichtig: Wanneer de geheugenlimiet wordt verlaagd, gaan alle VoIP Trace-gegevens verloren. Een back-up van de gegevens moet worden verzameld voordat het geheugen wordt verkleind.

Hoe VoIP-traceringsgegevens worden weergegeven

Om de gegevens van VoIP Trace weer te geven moeten we specifieke showopdrachten gebruiken. De gegevens kunnen worden weergegeven in dezelfde terminalsessie of kunnen ook worden verzonden via Syslog naar een off-box syslog server.

Opmerking: Sporen worden gedumpt na 32 seconden vanaf het moment dat een BYE wordt ontvangen voor een oproep.

Opmerking: De SIP-signalering wordt per been weergegeven en wordt niet gecombineerd als reguliere debugs. Regelmatige debugs zoals **debug cisco-berichten** tonen de SIP-signalering van een oproep in de exacte volgorde waarin de gebeurtenissen hebben plaatsgevonden. In VoIP Trace is elk been apart. Om de juiste volgorde te bepalen, worden de tijdstempels gebruikt.

De beschikbare opdrachten om de gegevens te tonen zijn:

```
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-spoor tonen

Deze opdracht geeft alle VoIP Trace-gegevens weer die in de buffer beschikbaar zijn. Het gebruik van deze opdracht heeft invloed op de prestaties van de router. Zodra de opdracht is ingevoerd, wordt een waarschuwingsbericht getoond om te waarschuwen voor het risico en te bevestigen om door te gaan:

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

dekkingbuffers voor voip-sporen tonen

Deze opdracht geeft een overzicht van de gespreksdetails voor alle oproepen die onder VoIP Trace worden gemeld. Elke call leg heeft een cover buffer gecreëerd die een samenvatting van de

geregistreerde vraag bevat.

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

Raadpleeg de volgende tabel voor meer informatie over elk veld:

Veld	Beschrijving
Zoektoets	Bevat een combinatie van bellen, nummer en call-id
tijdstempel	Tijd van aanmaken van afdekbuffer
Buffer-ID	Buffer-ID van de dekkingsbuffer
Bel-id	Nummerherkenning van de respectieve callpoot van de buffer naar de cover buffer
Peer-CallID	Bel-id van de peer leg
correlator	FPI-correlator van de oproep
Oproepnummer	Oproepnummer van de respectieve call-leg van de cover buffer
Telefoonnummer	Roepnummer van de respectieve call-leg van de cover buffer
SIP-gespreks-id	SIP-oproepnummer van de respectieve oproeppoot van de dekkingsbuffer
SIP-sessie-id	SIP-sessie-id van de respectieve call-leg van de cover buffer
GUID	GUID van de respectieve roep van de dekkingsbuffer
ankerbeen	De ankerpoot wordt op ja ingesteld als de respectieve aanroeppoot een ankerpoot is in de oproepvorkstroom of de media-proxyinzet
gevorkt been	Forked Leg is op ja ingesteld als de respectievelijke call-leg een ankerpoot is in de call forking flow of media proxy-implementatie
Bijbehorende Call-id's	Nummerherkenning van de bijbehorende gevorkte poten

Om de afdekbuffers te filteren kunnen we de opdrachten **Inclusief** en **sectie** gebruiken:

```
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
```

call-id voor voip-spoor tonen

In combinatie met de vorige opdracht, **tonen voip spoor call-id** kan worden gebruikt om de vraag te vinden. Nadat de call-id is geïdentificeerd, kan deze opdracht worden gebruikt om alle informatie over de specifieke call-leg weer te geven:

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

statistieken voip-spoor tonen

Deze show commando toont gedetailleerde uitvoer over status, geheugenverbruik, fouten of storingen, succesvolle oproepen, tijdstempels van nieuwste en oudste vermeldingen en meer.

```
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
```

Raadpleeg de volgende tabel voor meer informatie over elk veld:

Veld	Beschrijving
Overtrekstatus	Hier wordt de overtredingsstatus weergegeven, inclusief tijd en datum waarop VoIP-overtrekken is ingeschakeld.
Geheugenlimiet ingesteld	Geeft de ingestelde geheugenlimiet weer. Dit is 10% van het geheugen van de processorpool
Verbruikt geheugen	Geeft de hoeveelheid geheugen weer die dynamisch wordt verbruikt voor VoIP Trace
Totale gedumpte vraagbenen	Toont het aantal ontbroken vraagbenen die in logboekbuffer worden gedumpt. Gedumpte aanroepen verwijst naar aanroepbenen die geassocieerd zijn met IEC-fouten
oudste spoor gedumpt	Weergave tijdstempels en zoeksleutel van de oudste mislukte aanroep sinds VoIP Trace ingeschakeld was
Laatste spoor gedumpt	Weergave tijdstempels en zoeksleutel van de laatste mislukte aanroep sinds VoIP Trace ingeschakeld was
Totaal aantal opgenomen callbenen	Hiermee worden totale benen weergegeven die zijn opgenomen nadat VoIP Trace is ingeschakeld
Totale beschikbare vraagbenen	Hiermee worden in totaal beschikbare aanroepbenen in de geschiedenis weergegeve Dit kan hetzelfde of anders zijn dan Totale benen van de oproep die zijn opgenomen afhankelijk van de geheugenlimiet.
Oudste spoor beschikbaar	Toont tijdstempel en zoeksleutel van de oudste cover buffer beschikbaar in het gehe
Nieuwste spoor beschikbaar	Geeft tijdstempel en zoeksleutel weer van de nieuwste cover buffer beschikbaar in h geheugen
Totaal gemiste sporen	Het aantal gemiste vraagbenen van vertoningen wegens geheugengrens.

Aanvullende showopdrachten

Veld	Gebruik	
voip trace correlator tonen <correlator>	toon voip spoor correlator 4	Filters en displays VOIP Trace voor een specifieke call-id va
voip trace-id tonen <sessie-id>	spraaktracering sessie-id tonen 87003120822b5dbd8fd80f62d8e57c48	Filters en displays VOIP Trace voor e externe UUID van de sessie-ID-heade van de oproep weer te geven.
sip-call-id voor voip-trace tonen <call-id>	voip-spoor sip-call-id 10e60dfa9d8442848336d79e3155a8a1 weergeven	Filters en displays VOIP-tracering op

Over deze vertaling

Cisco heeft dit document vertaald via een combinatie van machine- en menselijke technologie om onze gebruikers wereldwijd ondersteuningscontent te bieden in hun eigen taal. Houd er rekening mee dat zelfs de beste machinevertaling niet net zo nauwkeurig is als die van een professionele vertaler. Cisco Systems, Inc. is niet aansprakelijk voor de nauwkeurigheid van deze vertalingen en raadt aan altijd het oorspronkelijke Engelstalige document (link) te raadplegen.