

Pesquise defeitos a falha da validação da capacidade do gateway da chamada da cortesia CVP (CCB)

Índice

[Introdução](#)

[Pré-requisitos](#)

[Requisitos](#)

[Componentes Utilizados](#)

[Informações de Apoio](#)

[Sintomas](#)

[Troubleshooting](#)

[Solução](#)

[Solução permanente](#)

[Solução do final de teste](#)

Introdução

Este documento descreve como pesquisar defeitos um problema portal da Voz de cliente (CVP) CCB quando o chamador não obtém uma oferta CCB porque a capacidade do gateway do tronco excedeu.

Pré-requisitos

Requisitos

A Cisco recomenda que você tenha conhecimento destes tópicos:

- CVP
- Chamada da cortesia de Cisco CVP

[Componentes Utilizados](#)

As informações neste documento são baseadas nestas versões de software:

- Server 10.5 CVP
- Empresa unificada do centro de contato (UCCE) 10.5

As informações neste documento foram criadas a partir de dispositivos em um ambiente de laboratório específico. Todos os dispositivos utilizados neste documento foram iniciados com uma configuração (padrão) inicial. Se a sua rede estiver ativa, certifique-se de que entende o impacto potencial de qualquer comando.

Informações de Apoio

Antes que o problema de potencialidade do gateway esteja pesquisado defeitos, é importante compreender o processo de validação do tronco no CCB. Basicamente, o processo determina primeiramente o número de atendimentos da tabela de **Callback_current** com **EventTypeID** dentro (21,22,23); Pendente, Inprogress, provisórios para gateways específicos e lugar.

Em segundo, da mesma tabela de **Callback_current**, determine, o número de atendimentos terminados com a causa conectada: **EventTypeID** = 24 (terminado), e **CauseID** = 27 (conectado).

Finalmente o processo adiciona estes dois valores e compara-os com o número de troncos configurados sob o serviço **Survivability.tcl**.

Se o resultado está sobre o ponto inicial dos troncos configurado, o processo envia para trás uma falha (retorno 1), se não envia para trás a aprovação (retorno 0).

Em resumo, a fórmula para validar os troncos usados para o CCB é:

Troncos CCB < (tabela de **Callback_current** com **EventTypeID** dentro (21,22,23); Pendente, Inprogress, provisórios para gateways específicos) + tabela de **Callback_current** de **EventTypeID** = 24 (terminado), e **CauseID** = 27 (conectado)

Se o valor dos troncos CCB é mais baixo a validação falha.

Sintomas

Uma chamada recebida não obtém a oferta CCB. O atendimento vai diretamente enfileirar de qualquer maneira o tempo de espera calculado (EWT)

Troubleshooting

Etapa 1. Recolha logs de atividade do aplicativo de **CallbackEntry** do server do linguagem de marcação extensível da Voz (VXML).

Etapa 2. Procure dentro dos logs de atividade por todo o atendimento onde a validação não é nenhuma:

```
Validate_02,data,result,none
```

Qual significa que a validação não passou. Obtenha o GUID para este atendimento. Filtre o atendimento pelo callid da atividade e procure um callid como este exemplo:

```
start,parameter,callid=BBBBAAAACCCDDDDDEEEEEFFFFAAAABBBB
```

Etapa 3. Recolha logs do relatório CVP para o server do relatório. Encontre o mesmo callid nos logs do relatório CVP.

```
ValidateHandler:ValidateHandler.exec: ValidateHandler GUID=BBBBAAAACCCDDDDDEEEEEFFFFAAAABBBB  
results:none validation status bitmask=0x00000103
```

Etapa 4. Converta o número do bitmask ao binário. Use uma calculadora do programador: **00010000011**

Etapa 5. Verifique o bitmask do guia de relatório CVP para ver se há tabelas CCB. Você deve ver que a validação falha devido a "EXCEED_CAPACITY_GW".

APROVAÇÃO 00000000 00000001
 00000000 00000010 ICM_NO_SCHEDULED_ALLOWED
 00000000 00000100 ICM_NO_PREEMPTIVE_ALLOWED
 00000000 00001000 NOT_IN_QUEUE
 00000000 00010000 TOD
 00000000 00100000 EWT
 00000000 01000000 PROBE_FAILED_NO_RESPONSE
 00000000 10000000 PROBE_FAILED_NO_CONFIG
 00000001 00000000 EXCEED_CAPACITY_GW
 00000010 00000000 EXCEED_CAPACITY_QUEUE

Nota: ICM_NO_SCHEDULED_ALLOWED e o bit APROVADO são ajustados sempre

Etapa 6. Reduza a edição para baixo a uma fila específica. Verifique o CCB Servlet do server do relatório CVP a fim determinar se há alguma fila específica onde o CCB não está oferecido. Abra um navegador da Web e datilografe-o.

<http:// {IP de servidor Address}:8000/cvp/CallbackServlet?method=Diag> do relatório

Este é um exemplo de uma fila onde o CCB seja oferecido:

Queue Name	ICM EWT	DQRate	Reconnect Time	SLA Time	Pending Callbacks	Percent Callbacks	<rw-ewt>
billing	2000	2000	30	60	0	0	2.67

Position	Number	Callback	Remaining Wait Time	In Queue State	GUID	Entrance Date	Retry	Validation Status	Start Date	Updated	Recording
1	""	false	1986	INQUEUE	9977E70000010000000000612C6C90A	10:26:53 09/01/2017	false	0x0003	10:26:53 09/01/2017	10:26:53 09/01/2017	""

Este é um exemplo de uma fila onde o CCB não seja oferecido

Queue Name	ICM EWT	DQRate	Reconnect Time	SLA Time	Pending Callbacks	Percent Callbacks	<rw-ewt>
billing	2000	2000	30	60	0	0	2.67

Etapa 7. Verifique se as filas são servidas por um gateway específico. Verifique a configuração de gateway (parâmetros de aplicativo do Survivability).

```

application
service new-call flash:bootstrap.vxml
!
service survivability flash:survivability.tcl
  paramspace callfeature med-inact-det enable
  param ccb id:10.201.198.21;loc:CALO;trunks:512
  
```

Etapa 8. Se a configuração está correta, verifique a informação armazenada na base de dados do servidor do relatório (Informix) para determinar o número de chama estes gateway e lugar

específicos. Você pode verificar pela identificação CCB (10.201.198.21 neste caso) ou pelo locattion (CALO neste exemplo).

Etapa 9. No server do relatório, alcance o base de dados Informix.

Abra uma alerta do CMD e datilografe-a: **dbacces**

Navegue à **conexão > conectam**

Selecione o **exemplo do cvp**

datilografe o **cvp_dbadmin** username

datilografe a senha

selecione o **base de dados de callback@cvp**

retire e navegue aos idiomas de consulta

Etapa 10. Execute a pergunta:

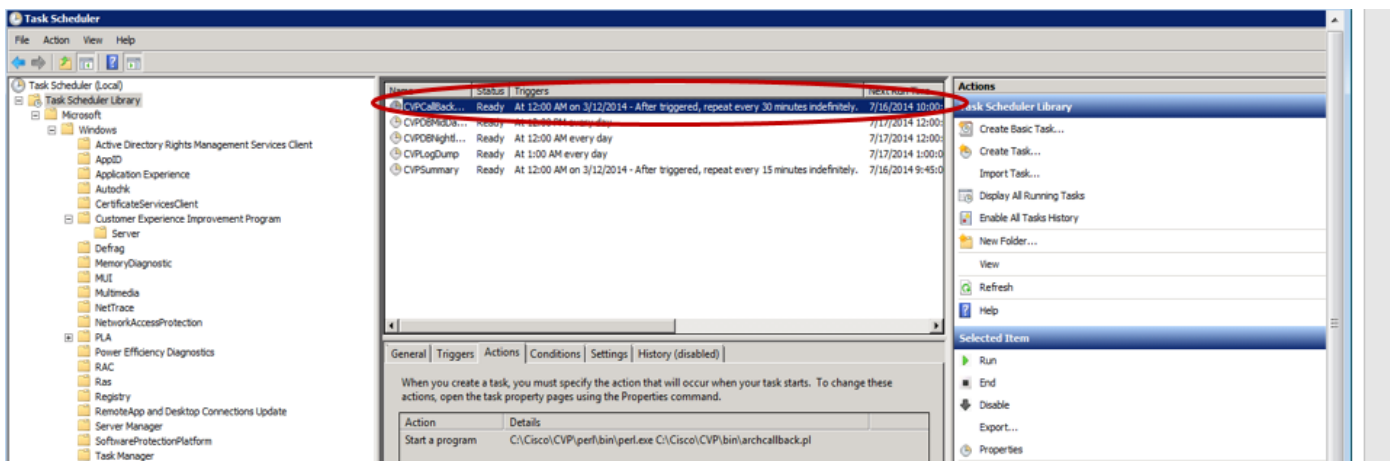
Selecione a contagem (*) de callback_current onde o == "CALO" do lugar;

Etapa 11. Se o valor é o mesmo ou mais altamente do que o valor do tronco configurado no gateway para os lugar, esta é a razão pela qual o validatidation falha, desde que os números máximos de troncos permitidos foram alcançados na tabela de Callback_Current.

Nota: Como provido no guia de relatório CVP, a tabela da rechamada é uma vista de duas tabelas: Callback_Current e Callback_Historical. As duas tabelas são idênticas. Cada 30 minutos, os dados para atendimentos terminados são puxados de Callback_Pending e movidos para Callback_Historical.

Etapa 12. Se o valor do tronco pelo lugar alcançou seus limites na tabela de Callback_Current e não há nenhuma rechamada na fila que este indica que há um problema em mover os registros da rechamada de Callback_Current para a tabela de Callback_Historical.

Etapa 13. Assegure-se de que CVPCallbackArchive esteja sendo executado sob as tarefas da programação (server do relatório CVP). Navegue ao **iniciar > programas > acessórios - > ferramentas de sistema - > tarefa programada.**



Etapa 14. Se esta tarefa **CVPCallbackArchive** termina assegure-se de que o código de saída esteja (0x0).

Name	Status	Triggers	Next Ru...	Last Run Ti...	Last Run Result	Author
CVPCallBack...	Ready	At 12:00 AM on 1/31/2017 -...	8/30/20...	8/30/2017 4...	The operation completed successfully. (0x0)	Administrator
CVPDBMidD...	Ready	At 12:00 PM every day	8/31/20...	8/30/2017 1...	The operation completed successfully. (0x0)	Administrator
CVPDBNight...	Ready	At 12:00 AM every day	8/31/20...	8/30/2017 1...	The operation completed successfully. (0x0)	Administrator
CVPLogDump	Ready	At 1:00 AM every day	8/31/20...	8/30/2017 1...	The operation completed successfully. (0x0)	Administrator
CVPSummary	Ready	At 12:00 AM on 1/31/2017 -...	8/30/20...	8/30/2017 4...	The operation completed successfully. (0x0)	Administrator
GoogleLeds	Ready	Multiple triggers defined	8/30/20...	8/30/2017 2...	The operation completed successfully. (0x0)	

Etapa 15. Se etapa 13 e 14 é muito bem, mas ainda nenhuns dados na tabela de **Callback_Historical**, você precisará de determinar porque a informação não é adicionada no base de dados. Verifique a integridade da informação armazenada na corrente e na tabela de histórico. Execute esta pergunta na janela CMD dos dbaccess do informix:

```
Select count (*) from callback_current where surrogateid in (select surrogateid from callback_historical);
```

Etapa 16. Se a contagem é 1 ou mais alta, significa que o chave principal na tabela atual já existe na tabela de histórico e a informação não está adicionada no base de dados. Na maioria destas encenações, uma race condition faz com que os registros duplicados participem na tabela **callback_current**.

O GUID ao mapeamento do surrogateid acontece na tabela da fila. Nas situações aonde o atendimento se move da espera da chamada para o script da fila da chamada, parece haver um indicador onde o trabalho do arquivo mova os registros da corrente para a história e o aplicativo incorpore um novo registro à tabela atual com o mesmo surrogateid. Esta edição é relacionada a este CDETS [CSCuq86400](#)

Solução

Etapa 1. Alcance o base de dados Informix. Abra uma alerta do CMD e datilografe-a: **dbaccess**

Etapa 2. Navegue à **conexão > conectam o exemplo seletor do cvp**. Datilografe o **cvp_dbadmin** username e datilografe a senha

Etapa 3. Selecione a saída do base de dados de **callback@cvp** e navegue aos **idiomas de consulta**

Etapa 4. Execute estes comandos:

supressão de callback_current onde surrogateid dentro (surrogateid seletor de callback_historical);

Se há um erro da tabela provisória faça:

deixe cair o T1 da tabela;

Etapa 5. Execute o procedimento sp que move a informação de atual para a tabela histórica da chamada dos dbaccess do indicador do idioma de consulta.

EXECUTE o sp_arch_callback() do PROCEDIMENTO;

Etapa 6. Certifique-se de não haja tantos como registros na tabela atual como antes.

Selecione a contagem (*) de callback_current onde o == "CALO" do lugar;

Solução permanente

Etapa 1. Navegue a **Cisco \ CVP \ informix_frag** e abra **sp_arch_callback.sql** em um editor de texto.

Etapa 2. Uncomment esta linha no início do arquivo: **--sp_arch_callback** do procedimento da **gota**; (remova -- no começo da linha).

Etapa 3. Adicionar esta linha: **supressão de callback_current onde surrogateid dentro (surrogateid seletor de callback_historical)**; em seguida

crie a linha do sp_arch_callback() do procedimento.

Etapa 4. Salvar o arquivo.

Etapa 5. Este é um exemplo em como o primeiro parte do arquivo deve olhar como.

```
{*****
Stored procedure to move completed calls out of the active table into the
historical table.
*****}
drop procedure sp_arch_callback;
create procedure sp_arch_callback()

DEFINE p_ageoff INTEGER;

-- delete any duplicates found in current table.

delete from callback_current where surrogateid in (select surrogateid from callback_historical);
```

Solução do final de teste

Etapa 1. Abra uma alerta do CMD e execute o comando: **dbschema**

dbschema - chamada d - sp_arch_callback f

Nota: Se você tem uma edição da autorização ao executar o comando do dbschema, o início de uma sessão como o cvp_dbadmin no server do relatório e a tentativa mais uma vez.

Etapa 2. Da saída, assegure-se de que a **supressão do comando** esteja executada.

```
C:\Users\Administrator>dbschema -d callback -f sp_arch_callback
DBSCHEMA Schema Utility          INFORMIX-SQL Version 12.10.FC3

create procedure "Administrator".sp_arch_callback()
DEFINE p_ageoff INTEGER;
-- delete any duplicates found in current table.
delete from callback_current where surrogateid in (select surrogateid from callb
ack_historical);
SELECT surrogateid
FROM Callback_current
WHERE EventTypeID in (24,29) -- Completed, Too many callbacks
AND CauseID in (27,28) -- Connected, Cancelled
INTO TEMP t1 WITH NO LOG;
```