Dépannez les échecs d'appel quand le service de capacité de survie est mis en application

Table des matières

Introduction

Ce document décrit comment dépanner un échec d'appel quand le script de capacité de survie du Customer Voice Portal (CVP) est configuré sur l'homologue de numérotation en entrée d'une passerelle d'entrée.

Contribué par Kabeer Noorudeen, ingénieur TAC Cisco.

Conditions préalables

Exigences

Cisco vous recommande de prendre connaissance des rubriques suivantes :

- Écoulement complet d'appel CVP
- Passerelle IOS et protocole PRI
- Intelligent Contact Management de Cisco Unified (missile aux performances améliorées), déploiements du Cisco Unified Contact Center Enterprise (UCCE)

Composants utilisés

Les informations contenues dans ce document sont basées sur les versions de logiciel suivantes :

- Serveur 9.0 CVP et en haut
- UCCE 9.0 et en haut
- Passerelle IOS 15.X

Les informations contenues dans ce document ont été créées à partir des périphériques d'un environnement de laboratoire spécifique. Tous les périphériques utilisés dans ce document ont démarré avec une configuration effacée (par défaut). Si votre réseau est opérationnel, assurez-vous que vous comprenez l'effet potentiel de toute commande.

Problème

Quand un service est ajouté à l'les pots entrants Dialpeer pour déclencher le script de capacité de survie, l'appel échoue avec cette cause :

ccCallDisconnect : Cause Value=81

Remarque: exécutez la commande de débogage, **debug voip ccapi inout**, afin de voir code d'erreur signalé quand l'appel échoue. En outre, exécutez la commande de débogage **mettent au point le message de ccsip** si vous voulez voir les messages d'erreur de SIP signalés quand l'appel échoue.

Des messages d'erreur contradictoires sont signalés quand le **message de ccsip de débogage** est activé.
Les messages SIP, ne donnent pas beaucoup d'indication autre que cela que l'appel échoue après que l'étiquette de VRU réseau soit retournée, qui peut être fallacieuse.

Si le service est enlevé du des pots entrants cadran-pair, tout fonctionne bien.

Procédure

Étape 1. On que les passerelles d'entrée activent met au point : **le debug voip ccapi inout**, **mettent au point le message de ccsip, mettent au point l'application vocale tous,** et le **debug isdn q931**.

Étape 2. Recréez le problème. Placez un appel à la passerelle.

Étape 3. Collectez les logs. Utilisez la commande, **show logging.**

Si vous regardez les logs CCAPI (**debug voip ccapi inout**), vous voyez juste que l'appel obtient avec succès remis outre du script de capacité de survie et alors l'appel échoue :

Spoiler

* 13 septembre 19:29:57.507 CT : //21/9E2AF1DC800C/CCAPI/cc_process_call_setup_ind :
>>>>CCAPI a remis le cid 21 avec la balise 101 à l'app « _ManagedAppProcess_cvp-survivability »
* 13 septembre 19:29:58.507 CT : //21/9E2AF1DC800C/CCAPI/ccCallSetupAck :
Appel Id=21

* 13 septembre 19:29:58.551 CT : //21/9E2AF1DC800C/CCAPI/cc_api_call_disconnected :
Cause Value=81, Interface=0x2500E10, appel Id=21
* 13 septembre 19:29:58.551 CT : //21/9E2AF1DC800C/CCAPI/cc_api_call_disconnected :
Entrée d'appel (Responsed=FALSE, cause Value=81, relance Count=0)
* 13 septembre 19:29:58.551 CT : //22/9E2AF1DC800C/CCAPI/ccCallDisconnect :
Cause Value=81, Tag=0x0, entrée d'appel (débranchement précédent Cause=0, débranchement Cause=0)
* 13 septembre 19:29:58.551 CT : //22/9E2AF1DC800C/CCAPI/ccCallDisconnect :
Cause Value=81, entrée d'appel (Responsed=FALSE, cause Value=81)
* 13 septembre 19:29:58.551 CT : //21/9E2AF1DC800C/CCAPI/ccCallDisconnect :
Cause Value=81, Tag=0x0, entrée d'appel (débranchement précédent Cause=0, débranchement Cause=81)
* 13 septembre 19:29:58.551 CT : //21/9E2AF1DC800C/CCAPI/ccCallDisconnect :
Cause Value=81, entrée d'appel (Responsed=TRUE, cause Value=81)

* 13 septembre 19:29:57.507 CT : //21/9E2AF1DC800C/CCAPI/cc_process_call_setup_ind : >>>>CCAPI a remis le cid 21 avec la balise 101 app « _ManagedAppProcess_cvp-survivability " * au 13 septembre 19:29:58.507 CT : //21/9E2AF1DC800C/CCAPI/ccCallSetupAck : Appel Id=21*Sep 13 19:29:58.551 CT : //21/9E2AF1DC800C/CCAPI/cc_api_call_disconnected : Cause Value=81, Interface=0x2500E10, appel Id=21*Sep 13 19:29:58.551 CT : //21/9E2AF1DC800C/CCAPI/cc_api_call_disconnected : Entrée d'appel (Responsed=FALSE, cause Value=81, relance Count=0)*Sep 13 19:29:58.551 CT : //22/9E2AF1DC800C/CCAPI/ccCallDisconnect : Cause Value=81, Tag=0x0, entrée d'appel (débranchement précédent Cause=0, débranchement Cause=0)*Sep 13 19:29:58.551 CT : //22/9E2AF1DC800C/CCAPI/ccCallDisconnect : Cause Value=81, entrée d'appel (Responsed=FALSE, cause Value=81)*Sep 13 19:29:58.551 CT : //21/9E2AF1DC800C/CCAPI/ccCallDisconnect : Cause Value=81, Tag=0x0, entrée d'appel (débranchement précédent Cause=0, débranchement Cause=81)*Sep 13 19:29:58.551 CT : //21/9E2AF1DC800C/CCAPI/ccCallDisconnect : Cause Value=81, entrée d'appel (Responsed=TRUE, cause Value=81)

Si vous regardez met au point pour TCL (par exemple **application toute de** debugvoice**),** vous voyez l'appelant déconnecté quand les essais TCL pour obtenir l'état d'appel en cours.

Spoiler

* 13 septembre 20:29:54.211 CT : //81//TCL : /tcl_InfotagObjCmd : l'infotag obtiennent evt_state_current

* 13 septembre 20:29:54.211 CT : //81//TCL : /tcl_InfotagGetObjCmd : l'infotag obtiennent evt_state_current
* 13 septembre 20:29:54.211 CT : //81//AFW_:/vtr_ev_state_current : argc 2
* 13 septembre 20:29:54.211 CT : //81//AFW_:/vtr_ev_state_current : Événement [CALL_INIT]
*** 13 septembre 20:29:54.211 CT : //81//TCL : /tcl_FSMObjCmd : setstate CALLER_DISCONNECTED FSM**
*** 13 septembre 20:29:54.211 CT : //81//TCL : /tcl_FSMSetStateObjCmd : setstate CALLER_DISCONNECTED de setstate**
* 13 septembre 20:29:54.211 CT : //81//TCL : /tcl_InfotagObjCmd : l'infotag obtiennent le con_ofleg 81
* 13 septembre 20:29:54.211 CT : //81//TCL : /tcl_InfotagGetObjCmd : l'infotag obtiennent le con_ofleg 81
* 13 septembre 20:29:54.211 CT : //81//AFW_:/vtr_co_ofleg : argindex 2 de l'argc 3
* 13 septembre 20:29:54.211 CT : //81//Tcl : /tcl_parseCallID_vartagObj : Tronçon Count=1 de traduction VARTAG
* 13 septembre 20:29:54.211 CT : //81//AFW_:/vtr_co_ofleg : EV_CONNECTIONS []
* 13 septembre 20:29:54.211 CT : //81//TCL : /tcl_LegObjCmd : débranchement 81 de tronçon
* 13 septembre 20:29:54.211 CT : //81//TCL : /tcl_LegDisconnectObjCmd : débranchement 81
* 13 septembre 20:29:54.211 CT : //81//Tcl : /tcl_parseCallID_vartagObj : Tronçon Count=1 de traduction VARTAG
* 13 septembre 20:29:54.211 CT : //81//TCL : /tcl_InfotagObjCmd : l'infotag obtiennent evt_state_current * 13 septembre 20:29:54.211 CT : //81//TCL : /tcl_InfotagGetObjCmd : l'infotag obtiennent evt_state_current * 13 septembre 20:29:54.211 CT : //81//AFW_:/vtr_ev_state_current : argc 2*Sep 13 20:29:54.211 CT : //81//AFW_:/vtr_ev_state_current : Événement [CALL_INIT] * 13 septembre 20:29:54.211 CT : //81//TCL : /tcl_FSMObjCmd : setstate CALLER_DISCONNECTED FSM * 13 septembre 20:29:54.211 CT : //81//TCL : /tcl_FSMSetStateObjCmd : setstate CALLER_DISCONNECTED de setstate * 13 septembre 20:29:54.211 CT : //81//TCL : /tcl_InfotagObjCmd : l'infotag obtiennent le con_ofleg 81 * 13 septembre 20:29:54.211 CT : //81//TCL : /tcl_InfotagGetObjCmd : l'infotag obtiennent le con_ofleg 81 * 13 septembre 20:29:54.211 CT : //81//AFW_:/vtr_co_ofleg : argindex 2*Sep 13 20:29:54.211 CT de l'argc 3 : //81//Tcl : /tcl_parseCallID_vartagObj : Tronçon Count=1*Sep 13 20:29:54.211 CT de traduction VARTAG : //81//AFW_:/vtr_co_ofleg : EV_CONNECTIONS [] * 13 septembre 20:29:54.211 CT : //81//TCL : /tcl_LegObjCmd : débranchement 81 de tronçon * 13 septembre 20:29:54.211 CT : //81//TCL : /tcl_LegDisconnectObjCmd : débranchement 81 * 13 septembre 20:29:54.211 CT : //81//Tcl : /tcl_parseCallID_vartagObj : Tronçon Count=1 de traduction VARTAG

Maintenant, si vous plongez vers le bas aux messages RNIS (**debug isdn q931**) vous voyez que les messages ne sont pas en phase avec les messages PSTN. Il y a un message d'INSTALLATION qui est livré dedans juste après l'installation et avant la démarche d'appel. Le script de capacité de survie ne peut pas manipuler cette situation.

Spoiler

* 13 septembre 19:53:31.763 CT : Le RNIS Se0/0/0:23 Q931 : RX < - INSTALLATION palladium = callref 8 = 0x1114
*** 13 septembre 19:53:31.767 CT : Le RNIS Se0/0/0:23 Q931 : RX < - INSTALLATION palladium = callref 8 = 0x1114**
*** 13 septembre 19:53:31.767 CT : Le RNIS Se0/0/0:23 ** ERREUR ** : L3_BadPeerMsg : callid 0x0 du Cr 0x9114 de l'événement 0x62**
* 13 septembre 19:53:31.767 CT : Le RNIS Se0/0/0:23 Q931 : TX - > RELEASE_COMP palladium = callref 8 = 0x9114

* 13 septembre 19:53:32.767 CT : Le RNIS Se0/0/0:23 Q931 : TX - > CALL_PROC palladium = callref 8 = 0x9114
* 13 septembre 19:53:32.767 CT : Le RNIS Se0/0/0:23 Q931 : TX - > CONNECTEZ palladium = le callref 8 = le 0x9114
* 13 septembre 19:53:32.803 CT : Le RNIS Se0/0/0:23 Q931 : RX < - RELEASE_COMP palladium = callref 8 = 0x1114
* 13 septembre 19:53:32.807 CT : Le RNIS Se0/0/0:23 Q931 : RX < - RELEASE_COMP palladium = callref 8 = 0x1114
*** 13 septembre 19:53:32.807 CT : Le RNIS Se0/0/0:23 ** ERREUR ** : L3_BadPeerMsg : callid 0x0 du Cr 0x9114 de l'événement 0x5A**

* 13 septembre 19:53:31.763 CT : Le RNIS Se0/0/0:23 Q931 : RX < - INSTALLEZ palladium = le callref 8 = le 0x1114 * le 13 septembre 19:53:31.767 CT : Le RNIS Se0/0/0:23 Q931 : RX < - INSTALLATION palladium = callref 8 = 0x1114 * 13 septembre 19:53:31.767 CT : Le RNIS Se0/0/0:23 ** ERREUR ** : L3_BadPeerMsg : callid 0x0*Sep 13 19:53:31.767 CT du Cr 0x9114 de l'événement 0x62 : Le RNIS Se0/0/0:23 Q931 : TX - > RELEASE_COMP palladium = callref 8 = 0x9114 * 13 septembre 19:53:32.767 CT : Le RNIS Se0/0/0:23 Q931 : TX - > CALL_PROC palladium = callref 8 = 0x9114 * 13 septembre 19:53:32.767 CT : Le RNIS Se0/0/0:23 Q931 : TX - > CONNECTEZ palladium = le callref 8 = le 0x9114*Sep 13 19:53:32.803 CT : Le RNIS Se0/0/0:23 Q931 : RX < - RELEASE_COMP palladium = callref 8 = 0x1114 * 13 septembre 19:53:32.807 CT : Le RNIS Se0/0/0:23 Q931 : RX < - RELEASE_COMP palladium = callref 8 = 0x1114 * 13 septembre 19:53:32.807 CT : Le RNIS Se0/0/0:23 ** ERREUR ** : L3_BadPeerMsg : callid 0x0 du Cr 0x9114 de l'événement 0x5A

Solution

Le L3_BadPeerMsg est la clé ici pour résoudre ce problème. Cette erreur est signalée quand il y a une non-concordance sur le type de commutateur entre la passerelle et le PSTN.

La recommandation est d'exécuter la commande de placer le type de commutateur RNIS que le PSTN et la passerelle conviennent. Dans ce scénario le type de commutateur sur le PSTN est primaire-**Ni.** Pour des clients des USA, le commutateur-type utilisé généralement est primaire-**Ni.**

Le primaire-Ni de commutateur-type de la commande le **RNIS** configuré sur la passerelle d'entrée a résolu le problème.