

CVP と UCCE 間のハートビートをトラッキングするプロシージャ

目次

[はじめに](#)

[前提条件](#)

[要件](#)

[使用するコンポーネント](#)

[背景説明](#)

[手順](#)

概要

この資料に VRU PIM ログおよび Wireshark キャプチャを使用して Cisco Unified Customer Voice Portal (CVP) Callserver と Voice Response Unit Peripheral Interface Manager (VRU PIM) 間のハートビート メッセージをトラッキングする方法を記述されています。

前提条件

要件

次の項目に関する知識が推奨されます。

- CVP Server
- Cisco Unified Intelligent Contact Management (ICM)、Cisco Unified Contact Center Enterprise (UCCE) 配備
- Callserver および Voice Response Unit Peripheral Interface Manager (VRU PIM)

使用するコンポーネント

このドキュメントの情報は、次のソフトウェアのバージョンに基づくものです。

- CVP サーバ 9.0 以降
- UCCE 9.0 以上に
- Wireshark

本書の情報は、特定のラボ環境にあるデバイスに基づいて作成されたものです。このドキュメントで使用するすべてのデバイスは、初期 (デフォルト) 設定の状態から起動しています。稼働中のネットワークで作業を行う場合、コマンドの影響について十分に理解したうえで作業してください。

背景説明

ハートビート メッセージは CVP コール サーバの ICM サブシステムで設定されるポートを使用して CVP コール サーバと VRU PIM の間で交換されます。デフォルトポート数は 5000 です。

手順

CVP と VRU PIM 間のハートビート メッセージをトラッキングするために、実行される必要があるいくつかのステップがあります。

ステップ 1 VRU PIM 側で、トレースのレベルを上げて下さい。既定のトレース レベルはハートビート メッセージを表示しません。

heartbeat トレースだけ (*heartbeat * /on) 有効に すれば、ログは CVP と VRU PIM 間の heartbeat intearcation を、見ますこれだけ示しません:

```
18:58:00:552 pg2A-pim1 トレース: PIMActiveHeartbeat
```

```
18:58:05:536 pg2A-pim1 トレース: PIMActiveHeartbeat
```

```
18:58:10:536 pg2A-pim1 トレース: PIMActiveHeartbeat
```

```
18:58:15:537 pg2A-pim1 トレース: PIMActiveHeartbeat
```

```
18:58:20:537 pg2A-pim1 トレース: PIMActiveHeartbeat
```

しかしトレース *heartbeat が* /on およびトレース *session * /on あればそれは実際に VRU PIM のメッセージがシーケンス番号によって記録 することを示します:

方法の例は VRU PIM トレースを有効に するここにあります:

```
C:\icm\ins98\PG1A\logfile>procmon ins98 pg1a pin2
18:35:56 Trace: EMT Creating Mutex Global\IMTConnect_DisconnectLock
>>>>trace *session* /on
>>>>trace *heartbeat* /on
>>>>
```

警告 : 本番 envorment のトレースのレベルをできます degradade システムのパフォーマンス上げて下さい。

完全な 1 つのハートビート交換 3 つのメッセージがあります。

```
18:59:05:538 pg2A-pim1 Trace: PG->VRU: Heartbeat Req (= Message Type 5); Message Length 4 bytes
Invoke ID: (2857109196) aa4c06cc
```

```
18:59:05:538 pg2A-pim1 Trace: VRU->PG: Heartbeat Conf (= Message Type 6); Message Length 4 bytes
Invoke ID: (2857109196) aa4c06cc
```

```
18:59:05:538 pg2A-pim1 Trace: PIMActiveHeartbeat
```

そして 5 数秒後:

```
18:59:10:538 pg2A-pim1 Trace: PG->VRU: Heartbeat Req (= Message Type 5); Message Length 4 bytes
```

Invoke ID: (2857109197) aa4c06cd

18:59:10:538 pg2A-pim1 Trace: VRU->PG: **Heartbeat Conf** (= Message Type 6); Message Length 4 bytes
Invoke ID: (2857109197) aa4c06cd

18:59:10:538 pg2A-pim1 Trace: **PIMActiveHeartbeat**

そして後 5 秒毎にそれ

18:59:15:538 pg2A-pim1 Trace: PG->VRU: **Heartbeat Req** (= Message Type 5); Message Length 4 bytes
Invoke ID: (2857109198) aa4c06ce

18:59:15:538 pg2A-pim1 Trace: VRU->PG: **Heartbeat Conf** (= Message Type 6); Message Length 4 bytes
Invoke ID: (2857109198) aa4c06ce

18:59:15:538 pg2A-pim1 Trace: **PIMActiveHeartbeat**

18:59:20:538 pg2A-pim1 Trace: PG->VRU: **Heartbeat Req** (= Message Type 5); Message Length 4 bytes
Invoke ID: (2857109199) aa4c06cf

18:59:20:538 pg2A-pim1 Trace: VRU->PG: **Heartbeat Conf** (= Message Type 6); Message Length 4 bytes
Invoke ID: (2857109199) aa4c06cf

18:59:20:538 pg2A-pim1 Trace: **PIMActiveHeartbeat**

ステップ 2 キャプチャ Wireshark トレース。

VRU PIM ログ メッセージは十分に一般的に 2 つのコンポーネント間の heartbeat 相互対話を確認することです。ただし、あるシナリオでは wireshark キャプチャは必要です。

CVP Callserver から奪取される wireshark トレースの断片はここにあります。

2 つのデバイスがポート 5000 で話しているので、トレースは `tcp.port==5000` によってフィルタリングされます

次に wireshark のハートビート パケットは 66 バイトのための (プッシュ、Ack) および Ack のための 54 から 60 バイトです。

CVP Callserver から VRU PIM に送られるプッシュ メッセージの実際のキープアライブ データは逆も同じだった 12 バイトであり。イメージに示すように

No.	Time	Source	Destination	Protocol	Length	Info
34	2012-12-01 18:59:05.96897	192.168.0.25	192.168.0.27	TCP	66	opencore > complex-main [PSH, ACK] Seq=3382468379 Ack=1858767277 win=65079 Len=12
35	2012-12-01 18:59:05.96897	192.168.0.27	192.168.0.25	TCP	66	complex-main > opencore [PSH, ACK] Seq=1858767277 Ack=3382468391 win=64887 Len=12
38	2012-12-01 18:59:07.645295	192.168.0.25	192.168.0.27	TCP	54	opencore > complex-main [ACK] Seq=3382468391 Ack=1858767289 win=65067 Len=0
83	2012-12-01 18:59:10.517089	192.168.0.25	192.168.0.27	TCP	66	opencore > complex-main [PSH, ACK] Seq=3382468391 Ack=1858767289 win=65067 Len=12
84	2012-12-01 18:59:10.517089	192.168.0.27	192.168.0.25	TCP	66	complex-main > opencore [PSH, ACK] Seq=1858767289 Ack=3382468403 win=64875 Len=12
87	2012-12-01 18:59:10.5174784	192.168.0.25	192.168.0.27	TCP	54	opencore > complex-main [ACK] Seq=3382468403 Ack=1858767301 win=65055 Len=0
114	2012-12-01 18:59:14.208964	192.168.0.27	192.168.0.25	TCP	150	complex-main > opencore [PSH, ACK] Seq=1858767301 Ack=3382468403 win=64875 Len=12
117	2012-12-01 18:59:14.796464	192.168.0.25	192.168.0.27	TCP	54	opencore > complex-main [ACK] Seq=3382468403 Ack=1858767397 win=64959 Len=0
148	2012-12-01 18:59:15.570899	192.168.0.25	192.168.0.27	TCP	66	opencore > complex-main [PSH, ACK] Seq=3382468403 Ack=1858767397 win=64959 Len=12
149	2012-12-01 18:59:15.537089	192.168.0.27	192.168.0.25	TCP	66	complex-main > opencore [PSH, ACK] Seq=1858767397 Ack=3382468415 win=64863 Len=12
152	2012-12-01 18:59:15.724081	192.168.0.25	192.168.0.27	TCP	54	opencore > complex-main [ACK] Seq=3382468415 Ack=1858767409 win=64947 Len=0
199	2012-12-01 18:59:20.17281	192.168.0.25	192.168.0.27	TCP	66	opencore > complex-main [PSH, ACK] Seq=3382468415 Ack=1858767409 win=64947 Len=12
200	2012-12-01 18:59:20.537281	192.168.0.27	192.168.0.25	TCP	66	complex-main > opencore [PSH, ACK] Seq=1858767409 Ack=3382468427 win=64851 Len=12
201	2012-12-01 18:59:20.731570	192.168.0.25	192.168.0.27	TCP	54	opencore > complex-main [ACK] Seq=3382468427 Ack=1858767421 win=64935 Len=0

1 パケットを必要として下さい。一例として、パケット 34 を選択して下さい。この VRU PIM で見られる最初ハートビートメッセージの timestamp(18:59:05) のパケット matches は先に記録します。

フレームのちょうどデータの部分を拡張して下さい:

```

# Frame 34: 66 bytes on wire (528 bits), 66 bytes captured (528 bits)
# Ethernet II, Src: Vmware_96:10:ea (00:50:56:96:10:ea), Dst: Vmware_96:6c:e1 (00:50:56:96:6c:e1)
# Internet Protocol Version 4, Src: 192.168.0.25 (192.168.0.25), Dst: 192.168.0.27 (192.168.0.27)
# Transmission Control Protocol, Src Port: opencore (4089), Dst Port: complex-main (5000), Seq: 3382468379, Ack: 1858767277, Len: 12
# Data (12 bytes)
  Data: 0000000400000005aa4c06cc
    [Length: 12]

```

呼び出を用いるデータ **aa4c06cc** 一致が VRU PIM トレースで番号が付くのを表示できます。

続くハートビートパケットにそれから数 **aa4c06cd** が、**aa4c06ce** 等あります。