UCサーバのゾンビ/非アクティブプロセスのトラブルシューティング

内容

概要

このドキュメントでは、Admin CLIを使用してログインした場合に、CUCM、IMnP、およびその他のCisco UC製品で見られるゾンビプロセスを使用する方法について説明します。

前提条件

要件

UCサーバのAdmin CLIの使用に関する知識があることが推奨されます。

- Cisco Unified Communications Manager（CUCM）
- Cisco Unified Instant Messaging and Presence Server(IMnP)
- Cisco Unity Connection Server(CUC)

使用するコンポーネント

このドキュメントの内容は、特定のソフトウェアやハードウェアのバージョンに限定されるものではありません。

このドキュメントの情報は、特定のラボ環境にあるデバイスに基づいて作成されました。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。本稼働中のネットワークでは、各コマンドによって起こる可能性がある影響を十分確認してください。

背景説明

ユニファイドコミュニケーションサーバは、基本的にLinux OSベースのアプリケーションです。Linux上でプロセスが停止した場合、そのプロセス記述子(PID)はメモリから即座に削除されるわけではなく、メモリ内に残り、メモリの量はごくわずかです。このプロセスは非アクティブなプロセスになり、プロセスの親に子プロセスが終了したことが通知されます。その後、親プロセスはdeadプロセスの終了ステータスを読み取り、それをメモリから完全に削除するはずです。wait()システムコールを使用して行うと、ゾンビプロセスがプロセステーブルから削除されます。これは、ゾンビプロセスを刈り取ることと呼ばれます。これは一般的に非常に迅速に行われるため、システム上にゾンビプロセスが蓄積しているとは思えません。

ただし、親プロセスがwait()シグナル呼び出しを行わないことがあり、子プロセスがクリーンアップされるまでメモリ内に留まります。つまり、ゾンビプロセスは実行が完了したプロセスですが、まだプロセステーブルにエントリがあります。親プロセスは引き続き子の終了ステータスを読み取る必要があります。

UCOS Admin CLIを使用したゾンビの確認

CLIから、**show process load**コマンドを使用して、ゾンビの存在を確認**できます**。

```
admin:show process load
admin:show process load
top - 08:43:47 up 48 days,  4:20,  1 user,  load average: 9.86, 6.17, 4.17
Tasks: 879 total,   1 running, 861 sleeping,   0 stopped,  17 zombie
Cpu(s): 40.9%us, 10.2%sy,  0.4%ni, 48.1%id,  0.1%wa,  0.0%hi,  0.2%si,  0.0%st
Mem:   8062228k total,  7888104k used,   174124k free,    78128k buffers
Swap:  4095996k total,  2891264k used,  1204732k free,  2368392k cached
  PID USER      PR  NI  VIRT  RES  SHR S %CPU %MEM    TIME+  COMMAND
20038 xcpuser   20   0  813m 627m  21m S 18.9  8.0   4756:46 jabberd
```

ゾンビの手動トラブルシューティング/クリア

前述のようにPIDを保持するために使用される小さなメモリとは別に、ゾンビプロセスはシステムリソースを使用しませんが、プロセスIDは保持します。UCサーバでは、システムに提供されるメモリが大きいため、ゾンビの存在による他のプロセスのPIDがシステムで不足する可能性は非常に少なくなります。

したがって、システム上にゾンビを残しておいて、次のシステムのリブート時に自動的に**消去されます**。

ただし、システム内のゾンビをクリアする必要がある場合は、特定のアクションを実行できます

適切なサービスの再起動

関連するプロセスを特定し、その結果、子プロセスをリークするサービスを特定する必要があります。

1. CLI出力から、**show process list**および**show process list detail**の出力を取り出します。

```
3476 admin:show process list detail
3477 PID ARGS
3478   PID  PPID USER     COMMAND
```

```
3765 admin:show process list
3766 PID ARGS
3767   PID COMMAND
```

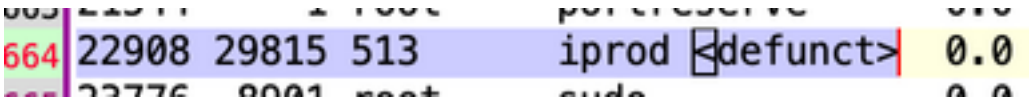
2. テキストエディタで出力をコピーし、ファイル内で「disct」というテキストを検索します。

3. これらの廃止されたプロセスのプロセスID(pid)と親プロセスID(ppid)をメモします。

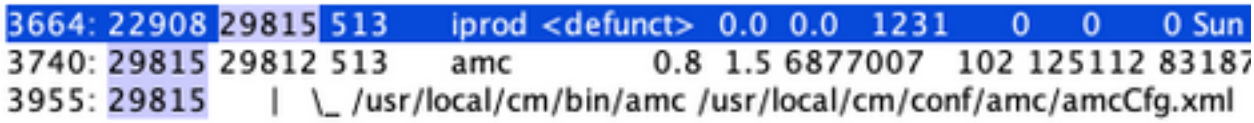
4. ドキュメント内のppidを追跡して、関連するプロセスを見つけます。

例 1

CUCM：ファイルを検索して「dect」というテキストを探すと、PID 22908が存在することが分かります。



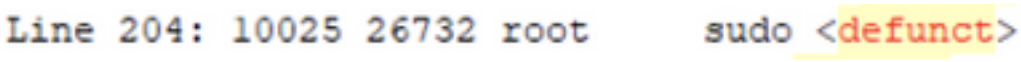
このPIDのppidは29815です。このドキュメントの29815を追跡すると、プロセスがAMCサービスに関連していることが分かります。



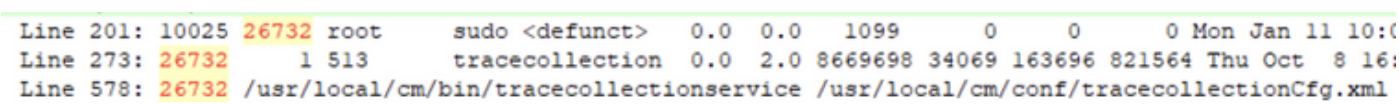
解決策：このノードでAMC(Alert Manager and Collector Service)を再起動すると、ゾンビがクリアされます。

例 2

CUCM：テキストがdectされた時、PID 10025が存在します。



このPIDのppidは26732です。ドキュメント内のトラッキング26732では、プロセスがTrace Collection Serviceに関連していることがわかります。



解決策：このノードでTrace Collection Serviceを再起動すると、ゾンビがクリアされます。

例 3

CUCM：テキストdectのファイルが検索されると、PID 23959が存在しなくなったことが分かります。

```
Line 252: 23959 26764 513      du <defunct>      0.0  0.0   3725
```

このPIDのppidは26764です。このドキュメントの26764を追跡すると、プロセスがCDR Repository service(cdrrep)に関連していることが分かります

```
Line 249: 23959 26764 513      du <defunct>      0.0  0.0   3725      1     0
Line 276: 26764     1 513      cdrrep            0.2  1.0 9631438 15471 80840 53681
Line 581: 26764 /usr/local/cm/bin/cdrrep /usr/local/cm/conf/cdrrep/cdrrepCfg.xml
```

解決策：CDRリポジトリサービスを再起動すると、このゾンビがクリアされます。

例 4

CUC：テキストが失効したファイルを検索すると、PID 325、370、387が3つ存在することがわかります。

```
: 325  7827 265    sftp <defunct>  0.0  0.0   940
: 370  7827 265    sftp <defunct>  0.0  0.0   943
: 387  7827 265    sftp <defunct>  0.0  0.0   943
```

これらすべてのPIDのppidは7827です。このドキュメントのトラッキング7827では、プロセスがConnection File Syncerサービスに関連していることがわかります。

```
415: 7827 /opt/cisco/connection/bin/CuFileSync
1509: 325  7827 265    sftp <defunct>  0.0  0.0
1511: 370  7827 265    sftp <defunct>  0.0  0.0
1513: 387  7827 265    sftp <defunct>  0.0  0.0
```

解決策 – 接続ファイル同期サービスを再起動すると、ゾンビがクリアされます。

例 5

IMnP：テキストdectのファイルが検索されると、PID 1806が存在しなくなったことが分かります。

```
1806  1775 sftpuser ssh <defunct>    0.0
```

そのPIDのppidは1775です。このドキュメントのトラッキング1775で、プロセスが同じクラスタ内の別のIMnPノードへのSFTP接続であることがわかります。

```
287: 1775 sftp sftpuser@imnpsub.emea.lab
511: 1775    1 sftpuser sftp          0.0  0.0   944
512: 1806  1775 sftpuser ssh <defunct>   0.0  0.0   1
```

解決策：IMnPでは、SFTPが所有する非アクティブなSSHプロセスが表示されることがあります。これらは表面的であることが判明し、サーバをリブートすると削除できます。

サーバをリブートします。

該当するサーバをリブートすると、プロセステーブル内のすべての古いエントリがクリアされ、その結果、システム内のゾンビがクリアされます。

親プロセスの強制終了

Linuxでは、ゾンビプロセスをSIGKILLシグナルを使って通常のプロセスが殺される方法で殺すことはできません。ただし、親プロセスは強制終了できます。このシナリオで使用するコマンドは次のとおりです。

kill -9 <ppid>

この回避策を実行するには、TACに連絡してください。重要なサービスが突然ダウンしないように、親プロセスを終了する際に注意してください。

確認

ゾンビがクリアされたら、同じコマンドshow process loadを使用してゾンビ数を確認します。

```
admin:show process load
top - 22:53:32 up 14 days,  8:33,  1 user,  load average: 0.26, 0.10, 0.02
Tasks: 256 total,   1 running, 255 sleeping,   0 stopped,   0 zombie
Cpu(s):  1.8%us,  1.2%sy,  0.0%ni, 96.9%id,  0.0%wa,  0.0%hi,  0.0%si,  0.0%st
Mem:   3925428k total,  3801924k used,   123504k free,   258168k buffers
```