

Troubleshoot Zombie/Defunct Processes in UC Servers

Contents

[Introduction](#)

[Prerequisites](#)

[Requirements](#)

[Components Used](#)

[Background Information](#)

[Check Zombies using UCOS Admin CLI](#)

[Troubleshoot/Clear the Zombies Manually](#)

[Restart the Appropriate Service](#)

[Reboot the Server](#)

[Kill the Parent Process](#)

[Verify](#)

Introduction

This document describes how to work with zombie processes seen on CUCM, IMnP, and other Cisco UC products when logged in using Admin CLI.

Prerequisites

Requirements

Cisco recommends that you have knowledge of using Admin CLI of the UC servers:

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

Components Used

This document is not restricted to specific software and hardware versions.

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. If your network is live, ensure that you understand the potential impact of any command.

Background Information

The Unified Communications servers are essentially Linux OS based applications. When a process dies on Linux, it is not all removed from memory immediately, its process descriptor (PID)

stays in memory which only takes a tiny amount of memory. This process becomes a defunct process and the process's parent is notified that its child process has died. The parent process is then supposed to read the dead process's exit status and completely remove it from the memory. Once this is done using the `wait()` system call, the zombie process is eliminated from the process table. This is known as reaping the zombie process. This generally happens very quickly, so you will not see zombie processes accumulating on your system.

However, sometimes the parent processes do not do the `wait()` signal call, and the child process will stick in the memory until cleaned up. In other words, a zombie process is a process whose execution is completed but it still has an entry in the process table, as the parent process still needs to read its child's exit status.

Check Zombies using UCOS Admin CLI

From the CLI, the presence of zombies can be checked using the command **show process load**.

```
14 admin:show process load
15 admin:show process load
16 top - 08:43:47 up 48 days, 4:20, 1 user, load average: 9.86, 6.17, 4.17
17 Tasks: 879 total, 1 running, 861 sleeping, 0 stopped, 17 zombie
18 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
19 Mem: 8062228k total, 7888104k used, 174124k free, 78128k buffers
20 Swap: 4095996k total, 2891264k used, 1204732k free, 2368392k cached
21 PID USER PR NI VIRT RES SHR S %CPU %MEM TIME+ COMMAND
22 20038 xcpuser 20 0 813m 627m 21m S 18.9 8.0 4756:46 iabherd
```

Troubleshoot/Clear the Zombies Manually

Apart from the tiny memory used to hold the PID as mentioned previously, Zombie processes do not use any system resources, but they do retain their process ID. In UC servers, the memory provided to the system is large so the possibility of the system to run out of PIDs for other processes due to the presence of Zombies is very less.

So, the zombies can be left on the system, where they are automatically cleared at the **next system reboot**.

However, if there is a requirement to clear out the zombies in the system, you can follow a certain line of action

Restart the Appropriate Service

It is required to figure out the concerned process and hence the service which leaks the child process.

1. From the CLI output, take the output of **show process list** and **show process list detail**.

```

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. Copy the outputs in a text editor and search the file for the text 'defunct'.
3. Note down the Process IDs (pid) and Parent Process IDs (ppid) for those defunct processes.
4. Track down the ppid in the document to find the associated process.

Example 1

CUCM: When I search the file for the text 'defunct', I see that there's a PID 22908 that is defunct.

```

664 22908 29815 513 iprod <defunct> 0.0

```

The ppid for that PID is 29815. On tracking 29815 in the document, I see that the process is related to AMC service.

```

3664: 22908 29815 513 iprod <defunct> 0.0 0.0 1231 0 0 0 Sun
3740: 29815 29812 513 amc 0.8 1.5 6877007 102 125112 83187
3955: 29815 | \_ /usr/local/cm/bin/amc /usr/local/cm/conf/amc/amcCfg.xml

```

Solution- Restart **AMC(Alert Manager and Collector Service)** on this node clears the Zombie.

Example 2

CUCM: When the file for the text **defunct** is searched, I see that there's a PID 10025 that is defunct.

```

Line 204: 10025 26732 root sudo <defunct>

```

The ppid for that PID is 26732. On tracking 26732 in the document, you see that the process is related to the Trace Collection Service.

```

Line 201: 10025 26732 root sudo <defunct> 0.0 0.0 1099 0 0 0 Mon Jan 11 10:0
Line 273: 26732 1 513 tracecollection 0.0 2.0 8669698 34069 163696 821564 Thu Oct 8 16:
Line 578: 26732 /usr/local/cm/bin/tracecollection /usr/local/cm/conf/tracecollectionCfg.xml

```

Solution -Restart the **Trace Collection Service** on this node clears the Zombie.

Example 3

CUCM: When the file for the text **defunct** is searched, you see that there's a PID 23959 that is defunct.

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

The ppid for that PID is 26764. On tracking 26764 in the document, I see that the process is related to the CDR Repository service(cdrrep)

```
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 53688
Line 581: 26764 /usr/local/cm/bin/cdrrep /usr/local/cm/conf/cdrrep/cdrrepCfg.xml
```

Solution - Restart the **CDR Repository Service** clears this Zombie.

Example 4

CUC: When the file for the text **defunct** is searched, you see that there's are three PIDs 325, 370, 387 that are defunct.

```
: 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
```

The ppid for all those PIDs is 7827. On tracking 7827 in the document, you see that the process is related to the Connection File Syncer service.

```
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
```

Solution - Restart the **Connection File Syncer** service clears the Zombies.

Example 5

IMnP: When the file for the text **defunct** is searched, you see that there's a PID 1806 that is defunct.

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

The ppid for that PID is 1775. On tracking 1775 in the document, you see that the process is an

SFTP connection to another IMnP node in the same cluster.

```
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
```

Solution - In IMnP, SFTP owned Defunct SSH processes might be seen. They have been found to be cosmetic and can be removed by a **reboot of the server**.

Reboot the Server

A reboot of the concerned server clears all the stale entries in the process-table and **consequentially clears the zombies in the system**.

Kill the Parent Process

From Linux, you can't kill zombie processes the way normal processes are killed with the SIGKILL signal — zombie processes are already dead. However, you can kill the parent process. The command used in that scenario are:

kill -9 <ppid>

Contact TAC for carrying out this workaround. Ensure care while killing the parent process to ensure that no critical service is abruptly brought down.

Verify

Once the zombies have been cleared, use the same command **show process load** to check the zombie count.

```
[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
```