

UC 서버에서 좀비/없어진 프로세스 트러블슈팅

목차

[소개](#)

[사전 요구 사항](#)

[요구 사항](#)

[사용되는 구성 요소](#)

[배경 정보](#)

[UCS Admin CLI를 사용하여 좀비 확인](#)

[수동으로 좀비 문제 해결/지우기](#)

[적절한 서비스 재시작](#)

[서버 재부팅](#)

[상위 프로세스 종료](#)

[다음을 확인합니다.](#)

소개

이 문서에서는 Admin CLI를 사용하여 로그인할 때 CUCM, IMnP 및 기타 Cisco UC 제품에 표시되는 좀비 프로세스를 사용하는 방법에 대해 설명합니다.

사전 요구 사항

요구 사항

Cisco에서는 UC 서버의 Admin CLI를 사용하는 방법에 대해 알고 있는 것이 좋습니다.

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

사용되는 구성 요소

이 문서는 특정 소프트웨어 및 하드웨어 버전으로 한정되지 않습니다.

이 문서의 정보는 특정 랩 환경의 디바이스를 토대로 작성되었습니다. 이 문서에 사용된 모든 디바이스는 초기화된(기본) 컨피그레이션으로 시작되었습니다. 네트워크가 작동 중인 경우 모든 명령의 잠재적인 영향을 이해해야 합니다.

배경 정보

Unified Communications 서버는 기본적으로 Linux OS 기반 애플리케이션입니다. Linux에서 프로세스가 중단되면 메모리에서 즉시 모두 제거되지는 않으며 PID(Process Descriptor)는 작은 양의 메모리만 사용하는 메모리에 유지됩니다. 이 프로세스는 사용되지 않는 프로세스가 되며 프로세스의 상위 프로세스가 하위 프로세스가 중단되었음을 알립니다. 그런 다음 상위 프로세스는 Dead 프로세스

의 종료 상태를 읽고 메모리에서 완전히 제거해야 합니다. wait() 시스템 호출을 사용하여 이 작업을 수행하면 프로세스 테이블에서 좀비 프로세스가 제거됩니다. 이것은 좀비 과정을 얻는 것으로 알려져 있습니다. 일반적으로 매우 빠르게 발생하므로 시스템에 좀비 프로세스가 누적되는 것을 볼 수 없습니다.

그러나 부모 프로세스에서 wait() 신호 호출을 수행하지 않는 경우가 있으며, 하위 프로세스는 정리 될 때까지 메모리에 유지됩니다. 즉, 좀비 프로세스는 실행이 완료되었지만 상위 프로세스가 자녀의 종료 상태를 읽어야 하므로 프로세스 테이블에 여전히 항목이 있는 프로세스입니다.

UCS Admin CLI를 사용하여 좀비 확인

CLI에서 show process load 명령을 사용하여 좀비의 존재를 확인할 수 있습니다.

```
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
```

수동으로 좀비 문제 해결/지우기

앞서 언급한 것처럼 PID를 유지하는 데 사용된 작은 메모리 외에도, 좀비 프로세스는 시스템 리소스를 사용하지 않지만 프로세스 ID를 유지합니다. UC 서버에서는 시스템에 제공된 메모리가 크기 때문에 좀비의 존재로 인해 시스템이 다른 프로세스에 대해 PID가 부족할 가능성이 매우 적습니다.

즉, 좀비는 시스템에 남겨둘 수 있고, 다음 시스템 재부팅에 자동으로 지워집니다.

그러나, 시스템에 좀비를 제거하도록 요구사항이 있다면, 여러분은 특정한 행동라인을 따를 수 있다

적절한 서비스 재시작

관련 프로세스를 파악하고 하위 프로세스를 유출하는 서비스를 파악해야 합니다.

1. CLI 출력에서 show process list의 출력과 show process list detail을 가져옵니다.

```
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. 텍스트 편집기의 출력을 복사하고 'defunct' 텍스트에 대한 파일을 검색합니다.
3. 존재하지 않는 프로세스에 대한 프로세스 ID(pid) 및 상위 프로세스 ID(ppid)를 기록해 둡니다.
4. 문서의 ppid를 추적하여 연관된 프로세스를 찾습니다.

예 1

CUCM:'작동 안 됨'이라는 문자에 대한 파일을 검색했더니 없어진 PID 22908이 있는 것 같습니다.

```
664 22908 29815 513 iprod <defunct> 0.0
```

해당 PID의 ppid는 29815입니다. 문서에서 29815를 추적하면 프로세스가 AMC 서비스와 관련된 것으로 보입니다.

```
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
```

해결 방법 - 이 노드에서 AMC(Alert Manager 및 Collector Service)를 재시작하면 Zombie가 지워집니다.

예: 2:

CUCM:그 텍스트 파일을 검색하면 없어진 PID 10025가 있는 것 같습니다.

```
Line 204: 10025 26732 root sudo <defunct>
```

해당 PID의 ppid는 26732입니다. 문서에서 26732를 추적하면 프로세스가 추적 수집 서비스와 관련된 것을 알 수 있습니다.

```
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
```

해결 방법 - 이 노드에서 추적 모음 서비스를 다시 시작하면 좀비가 지워집니다.

예: 3:

CUCM: 없어진 텍스트 파일이 검색되면, 없어진 PID 23959가 있는 것을 볼 수 있습니다.

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

해당 PID의 ppid는 26764입니다. 문서에서 26764를 추적하면 프로세스가 CDRREP(CDR 저장소 서비스)와 관련되어 있음을 알 수 있습니다.

```
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 5368
Line 581: 26764 /usr/local/cm/bin/cdrrep /usr/local/cm/conf/cdrrep/cdrrepCfg.xml
```

해결 방법 - CDR 저장소 서비스를 다시 시작하면 이 좀비가 지워집니다.

예 4

CUC: **없어진** 텍스트 파일을 검색하면, 325, 370, 387개의 PID가 존재하지 않는 것을 볼 수 있습니다.

```
: 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
```

모든 PID의 ppid는 7827입니다. 문서에서 7827을 추적하면 이 프로세스가 연결 파일 동기화 서비스와 관련이 있음을 알 수 있습니다.

```
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
```

해결 방법 - Restart the **Connection File Syncer** service(연결 파일 동기화 서비스를 재시작하면 좀비가 지워집니다.)

예 5

IMnP: **없어진** 텍스트 파일을 검색하면, **없어진** PID 1806이 있는 것을 볼 수 있습니다.

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

해당 PID의 ppid는 1775입니다. 문서의 추적 1775에서 프로세스가 동일한 클러스터의 다른 IMnP 노드에 대한 SFTP 연결임을 확인할 수 있습니다.

```
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
```

해결 방법 - IMnP에서 SFTP가 소유한 SSH 프로세스가 표시될 수 있습니다.서버를 재부팅하면 외관이 되고 제거할 수 있습니다.

서버 재부팅

관련 서버를 재부팅하면 프로세스 테이블에 있는 모든 오래된 항목이 지워지고 시스템에서 **좀비**가 지워집니다.

상위 프로세스 종료

Linux에서는 SIGKILL 신호를 통해 정상적인 프로세스가 처리되는 방식으로 좀비 프로세스를 제거할 수 없습니다. 좀비 프로세스는 이미 죽은 상태입니다.그러나 상위 프로세스를 종료할 수 있습니다.이 시나리오에서 사용되는 명령은 다음과 같습니다.

kill -9 <ppid>

이 해결 방법을 수행하려면 TAC에 문의하십시오.중요한 서비스가 갑자기 중단되지 않도록 상위 프로세스를 종료하면서 주의를 기울여야 합니다.

다음을 확인합니다.

좀비가 제거되었으면, 같은 명령 **show process load**를 사용하여 좀비 수를 확인합니다.

```
[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
```