

# StarOS 시설 충돌 트러블슈팅

## 목차

- [소개](#)
- [개요](#)
- [충돌 시나리오](#)
- [추락 원인](#)
- [다양한 유형의 충돌](#)
- [초기 로그 요구 사항](#)
- [분석 단계](#)
- [세션 복구](#)

## 소개

이 문서에서는 StarOs 기능 충돌을 찾아 해결하는 방법에 대해 설명합니다.

## 개요

경우에 따라 시스템 로직이 실패하여 소프트웨어 작업이 다시 시작되어 적절한 기능을 복원할 수 있습니다. 이로 인해 프로세스 충돌이 발생할 수 있습니다. 작업 시설 충돌은 StarOS에서 자주 보고되며 충돌의 근본 원인을 기반으로 필요한 조치를 취할 수 있습니다. 노드의 충돌을 식별하려면 다음 CLI 명령을 사용할 수 있습니다.

```
***** show crash list *****
Saturday April 15 05:05:56 SAST 2023
=====
#           Time           Process  Card/CPU/      SW           HW_SER_NUM
           Time           Process  PID            VERSION      CF / Crash Card
=====
1  2022-Dec-02+14:08:46  confdmgr 02/0/19342  21.26.13     NA
2  2022-Dec-02+14:48:08  confdmgr 02/0/31546  21.26.13     NA
3  2022-Dec-04+19:10:50  sessmgr  03/0/12321  21.26.13     NA
4  2022-Dec-21+03:34:13  sessmgr  04/0/12586  21.26.13     NA
```

유사한 충돌이 하나의 레코드로 통합됩니다. 레코드에는 이 충돌 유형이 발생한 횟수가 표시됩니다

```
***** CRASH #02 *****
SW Version      : 21.26.13
Similar Crash Count : 33 >>>>
Time of First Crash : 2022-Dec-02+14:10:05
```

```
Assertion failure at confdmgr/src/confdmgr_fsm.c:870
Note: State machine failure, state = 3
Function: confdmgr_fsm_state_wait_p0_handler()
Expression: 0
Code: CRASH
Procllet: confdmgr (f=1900,i=0)
Process: card=2 cpu=0 arch=X pid=31546 argv0=confdmgr
```

수신 show snmp trap history verbose 출력에서 일부 프로세스가 crash했음을 보여줍니다.

```
Fri Dec 26 08:32:20 2014 Internal trap notification 73 (ManagerFailure) facility sessmgr
instance 188 card 7 cpu 0
Fri Dec 26 08:32:20 2014 Internal trap notification 150 (TaskFailed) facility sessmgr instance
188 on card 7 cpu 0
Fri Dec 26 08:32:23 2014 Internal trap notification 1099 (ManagerRestart) facility sessmgr
instance 139 card 4 cpu 1
Fri Dec 26 08:32:23 2014 Internal trap notification 151 (TaskRestart) facility sessmgr
instance 139 on card 4 cpu 1
```

## 충돌 시나리오

충돌의 원인은 여러 가지가 있을 수 있습니다.

1. 다양한 통화 흐름 시나리오
2. 메모리 문제
3. 컨피그레이션 문제
4. 하드웨어 장애

## 추락 원인

StarOS에는 여러 작업 기능이 있으므로 기능에 따라 해당 기능이 개별적으로 작동하므로, 해당 기능에 문제가 있는 입력이 발생할 때마다 해당 오류 상태에서 복구하기 위해 협업공간을 충돌시킵니다.

## 다양한 유형의 충돌

1. 어설션 오류

```
***** CRASH #22 *****
SW Version      : 21.26.13
Similar Crash Count : 33
Time of First Crash : 2023-Apr-12+22:40:01
```

```
Assertion failure at sess/smgr/sessmgr_snx.c:9568 >>>>
Function: sessmgr_snx_send_drop_call()
Expression: result == SN_STATUS_SUCCESS
Proclet: sessmgr (f=87000,i=261)
Process: card=5 cpu=0 arch=X pid=12724 cpu=~0% argv0=sessmgr
```

## 2. 세그먼테이션 장애

```
***** CRASH #69 *****
SW Version : 21.13.3
Similar Crash Count : 2
Time of First Crash : 2019-Nov-25+07:53:54
Fatal Signal 11: Segmentation fault >>>>
Faulty address: 0x7ff6b4801036
Signal from: kernel
Signal detail: address not mapped to object
Process: card=8 cpu=1 arch=X pid=7316 argv0=vpp
Crash time: 2020-Feb-11+04:04:23 UTC
Build_number:
```

## 3. 치명적 신호

```
***** CRASH #01 *****
SW Version : 21.23.12
Similar Crash Count : 2
Time of First Crash : 2023-Jan-27+05:22:46

Fatal Signal 11: 11 >>>>>
PC: [04be6859/X] sessmgr_pgw_create_bearers()
Faulty address: 0x297116e4
Signal from: kernel
Signal detail: address not mapped to object
Process: card=9 cpu=1 arch=X pid=10383 cpu=~8% argv0=sessmgr
```

## 초기 로그 요구 사항

충돌 로그는 충돌 이벤트 정보의 중요한 소스 역할을 합니다. 소프트웨어 충돌이 발생하면 StarOS는 충돌 원인을 파악하는 데 도움이 될 수 있는 관련 데이터를 캡처하고 저장합니다. 이 정보는 시스템 메모리에 저장하거나 네트워크 서버에 전송 및 저장할 수 있습니다.

Core File(코어 파일) 또는 Mini Core File(미니 코어 파일): 코어 파일은 충돌이 발생한 PID에 해당합니다. 코어 파일의 이름은 "crash-<card no>-<cpu>-<pid>-<unixtime>-core" 형식으로 지정됩니다. 이 정보는 "show crash list" 명령 출력에서 찾을 수 있습니다.

Minicore 파일: 이 파일에는 현재 스택 추적, 과거 프로파일러 샘플, 과거 메모리 활동 샘플 및 기타 번들된 데이터를 비롯한 실패한 작업에 대한 정보가 독점 파일 형식으로 포함되어 있습니다.

코어 덤프(또는 전체 코어): 코어 덤프는 충돌이 발생한 직후 프로세스의 전체 메모리 덤프를 제공합니다. 이러한 메모리 덤프는 소프트웨어 충돌의 근본 원인을 파악하는 데 필수적인 경우가 많습니다.

충돌 서명: 충돌 서명은 공유 SSD(Show Support Details) 또는 기타 관련 소스에서 검토할 수 있습니다.

```
***** show crash list *****
Saturday April 15 05:05:56 SAST 2023
=====
#           Time           Process  Card/CPU/      SW           HW_SER_NUM
          CF / Crash Card
=====
1  2022-Dec-02+14:08:46  confdmgr  02/0/19342  21.26.13      NA
2  2022-Dec-02+14:48:08  confdmgr  02/0/31546  21.26.13      NA
3  2022-Dec-04+19:10:50  sessmgr   03/0/12321  21.26.13      NA
```

이제 crash 1에 대한 시그니처를 알고 싶다면 SSD에서 CRASH #01을 검색하거나 CLI에서 show crash number 1을 사용합니다.

From SSD

```
***** CRASH #01 *****
SW Version      : 21.26.13
Similar Crash Count : 1
Time of First Crash : 2022-Dec-02+14:08:46
```

```
Assertion failure at confdmgr/src/confdmgr_fsm.c:758
Note: State machine failure, state = 5
Function: confdmgr_fsm_state_wait_p1_handler()
Expression: 0
Code: CRASH
```

Using CLI

```
[local]abc# show crash number 1
Friday June 09 06:41:53 CDT 2023
***** CRASH #01 *****
SW Version      : 21.12.20.77760
Similar Crash Count : 1
Time of First Crash : 2021-Mar-31+15:58:06
```

```
Fatal Signal 6: Aborted
PC: [ffffe430/X] __kernel_vsyscall()
Note: User-initiated state dump w/core.
Signal from: sitmain pid=6999 uid=0
Process: card=9 cpu=0 arch=X pid=9495 cpu=~0% ar
```

문제가 발생한 특정 타임스탬프에서 SSD(Show Support Details) 및 syslogs를 검토합니다.

## 분석 단계

1. 크래시 스택/서명을 확인하고 해당 크래시 서명에 대한 버그가 있는지 확인해야 합니다.
2. 백트레이스를 분석하고 해당 시설이 어떤 기능을 망가뜨렸는지에 대한 단서를 얻기 위해 코어파일/미니코어를 분석해야 합니다.
3. 코어파일 디버깅이 완료되면 소프트웨어 결함 및 유사한 크래시 시그니처 및 백트레이스에 대한 기존 소프트웨어 결함이 있는지 여부를 확인해야 합니다.

## 세션 복구

StarOs 소프트웨어는 예측된 조건/이벤트 및 예측되지 않은 조건/이벤트를 모두 처리하도록 설계되었습니다. Cisco는 완벽한 소프트웨어를 갖추기 위해 노력하고 있지만, 불가피하게 실수가 발생하고 충돌이 발생할 수 있습니다. 그렇기 때문에 세션 복구 기능이 매우 중요합니다.

세션 복구 기능은 시스템 내에서 하드웨어 또는 소프트웨어 장애가 발생할 경우 가입자 세션 정보의 원활한 장애 조치 및 재구성을 통해 완전히 연결된 사용자 세션의 연결을 차단합니다. 세션 복구는 시스템 내의 주요 소프트웨어 프로세스(예: 세션 관리자 및 AAA 관리자)를 미러링하여 수행됩니다. 이러한 미러링된 프로세스는 유휴 상태(대기 모드)로 유지되며, 소프트웨어 장애가 발생할 경우(예: 세션 관리자 작업이 중단됨) 필요할 때까지 처리를 수행하지 않습니다.

Demux 작업, AAA 관리자 및 VPN 관리자와 같은 작업에는 특히 가입자 정보를 처리하기 위한 자동 복구 메커니즘이 시스템에 내장되어 있습니다. 세션 복구는 기본적으로 sessmgr 작업에 오류가 있거나 카드 수준 오류가 발생하여 통화 손실 없이 세션을 복구해야 하는 경우를 의미합니다.

- 시스템에서 대기 세션 관리자는 각 처리 카드에서 활성 상태이며 장애 발생 시 기본 세션 관리자로 신속하게 인계될 수 있습니다. 그런 다음 새 대기 ssmgr이 처리 카드에 생성됩니다.
- sessmgr 프로세스가 예기치 않게 실패할 경우 대기 sessmgr은 aaamgr(AAA 관리자)에서 백업된 정보를 검색하고 해당 세션을 재작성합니다.
- aaamgr이 실패하면 대기 aamgr은 sessmgr을 쿼리하여 관련 가입자 정보를 동기화합니다.
- demux-mgr 오류가 발생할 경우 모든 세션을 쿼리하고 통화 배포 정보 데이터베이스를 재구성합니다.
- 카드 레벨 이중화를 보장하기 위해 하나의 처리 카드는 하드웨어 또는 소프트웨어 장애가 발생할 경우 즉시 인계받을 수 있는 대기 카드 역할을 합니다. 그런 다음 스탠바이 카드는 다른 카드에서 실행 중인 피어 aaamgr에서 세션을 복구합니다.

```
***** show session recovery status verbose *****
Saturday April 15 05:11:17 SAST 2023
Session Recovery Status:
  Overall Status      : Ready For Recovery >>>>
  Last Status Update  : 5 seconds ago

cpu state      ----sessmgr---  ----aaamgr----  demux
active standby active standby active status
-----
```

3/0 Active	40	1	40	1	0	Good
4/0 Active	40	1	40	1	0	Good
5/0 Active	40	1	40	1	0	Good
6/0 Active	40	1	40	1	0	Good
7/0 Active	0	0	0	0	10	Good (Demux)
8/0 Active	40	1	40	1	0	Good
9/0 Active	40	1	40	1	0	Good
10/0 Active	40	1	40	1	0	Good
11/0 Standby	0	40	0	40	0	Good

이 번역에 관하여

Cisco는 전 세계 사용자에게 다양한 언어로 지원 콘텐츠를 제공하기 위해 기계 번역 기술과 수작업 번역을 병행하여 이 문서를 번역했습니다. 아무리 품질이 높은 기계 번역이라도 전문 번역가의 번역 결과물만큼 정확하지는 않습니다. Cisco Systems, Inc.는 이 같은 번역에 대해 어떠한 책임도 지지 않으며 항상 원본 영문 문서(링크 제공됨)를 참조할 것을 권장합니다.