Understand Graceful Assert Handling in StarOS S4-SGSN

Contents

Introduction

Prerequisites

- Requirements
- Components Used

Background Information

Problem

Solution

Configure

Verify

Troubleshoot

Introduction

This document describes the “Graceful Assert Handling” feature, introduced in release 21.5.0 of StarOS.

Prerequisites

Requirements

Cisco recommends that you have knowledge of these topics:

- StarOs
- Serving GPRS Support Node (SGSN)

Components Used

The information in this document is based on StarOS R21.5 and later.

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 feature can be found in the official documentation here: SGSN-Admin guide.

The Graceful Assert Handling framework enables graceful handling of subscriber sessions for which the ASSERT condition is hit at the time of call execution. This is achieved without affecting other subscriber sessions on the same proclet.

Normally, when the ASSERT condition is hit, the Session Manager (SessMgr) proclet restarts and recovers

all the subscriber sessions from the AAA Manager (AAAMgr). The recovered subscriber sessions are moved to the IDLE state.

When Graceful Assert Handling is enabled, the SessMgr proclet is not restarted. Instead, the SessMgr proclet recovers only the affected subscriber's session from the AAAMgr and clears the existing subscriber's session on the SessMgr. The recovered subscriber sessions are moved to the IDLE state. At the time of the recovery procedure, all messages directed towards the subscriber are dropped. After recovery, the subscriber continues to handle messages directed towards it. With this procedure, the subscriber sessions that remain on the SessMgr remain unaffected.

Problem

There are some corner and/or collision cases for which either the root cause fix is complex or root cause is unknown. In these cases, graceful assert approach is taken in order to avoid full session manager restart.

Solution

With graceful assert, you can clean up and restore the 1 session that hits the graceful assert condition.

There is no impact to any other session on the same sessmgr.

There is no SNMP trap or syslog for a graceful restart.

There is no KPI loss in case of a graceful assert. The task itself is not restarted.

However, the graceful asserts are handled like any other crash, meaning you get an entry in show crash list.

How to identify a graceful assert from the SSD:

- System-initiated state dump w/core. – is seen under "show crash" output before stack
- crashed proclet is either user-initiated or non-boxer – is seen after the stack under "debug console cpu" output
- pid 7939 facility sessmgr failover 5132->94 – under "debug console CPU" is not logged/seen in case of Graceful Assert

Configure

Graceful Assert Handling can be configured as follows:

```
configure
   debug controlled-assert s4sgsn
      [ disable | enable ] core-generation
      limit-per-assert  assert_value
      [ no ] test file-name  file_name line-number  line_num [ sequence-number  seq_num ]
      end
```

Take note:

- controlled-assert: Configures the controlled assert framework.
- s4sgsn : Configures the S4-SGSN controlled assert.

- core-generation: Configures core generation for controlled assert. Default: Enabled.
- limit-per-assert: Configures the limit per assert for controlled assert. Default: 5.
- test file-name file_name line-number line_num [sequence-number seq_num]: Configures controlled assert test handling.
    - file-name file_name: Configures the file name where assert control is required. file_name must be an alphanumeric string of 1 through 254 characters.
    - line-number line_num: Configures the line number where assert control is required. line_num must be an integer from 1 to 4294967295.
    - sequence-number seq_num: Configures the sequence number where assert control is required. seq_num must be an integer from 1 to 100. Default: 1.
- disable: Disables the specified action for a controlled assert framework.
- enable: Enables the specified action for a controlled assert framework.
- no: Removes the specified test configuration related to the controlled assert framework.

Example

```
******************** CRASH #93 ***********************
SW Version          : 21.5.19
Similar Crash Count : 8
Time of First Crash : 2019-May-21+06:57:14

Fatal Signal 6: Aborted
  Note: System-initiated state dump w/core.           <<< This note indicates a graceful assert.
  Process: card=10 cpu=0 arch=X pid=11573 cpu=~16% argv0=sessmgr
  Crash time: 2019-May-23+06:00:13 UTC
  Recent errno: 11 Resource temporarily unavailable
  Build_number: 71813
```

Verify

Use this section to confirm that your configuration works properly.

Example of getting the controlled assert stats for all active sessmgrs:

```
# zcat ssd_s4sgn.log.gz | sed -n -e '/\*\{7\} show session subsystem facility sessmgr all debug-info /,/\*\{7\}/p' | sed -e
```

Example output:

```
SessMgr:  Instance 135
Controlled Assert Stats
   Module Name :SGW_DRV
   Assert Count:0
```

```
 Count    File:Line                                                    Last Assert hit time(in sec)

 Module Name :S4_SGSN
Assert Count:1
 Count    File:Line                                                    Last Assert hit time(in sec)
   1        sess/sgsn/sgsn-app/s4_sm/s4_smn_egtpc.c:3164                 2019/01/30 09:28:11 UTC
```

This information (count and line number) is reset if the sessmgr goes for a restart for any other crash. After the maximum number of times (default 5) is reached, the core is not generated.

Troubleshoot

There is currently no specific troubleshooting information available for this configuration.