Line Cards Power Down due to Communication Failures Troubleshooting Guide

Document ID: 116115

Contributed by Shashank Singh and Saurav Lahiri, Cisco TAC Engineers.
Jun 20, 2013

Contents

Introduction

This document describes how to troubleshoot line cards that have powered down because of communication failures on Cisco Catalyst 6500 Series Switches.

Prerequisites

Requirements

There are no specific requirements for this document.

Components Used

The information in this document is based on the Cisco Catalyst 6500 Series Switches and is not limited to a specific software version.

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, make sure that you understand the potential impact of any command.

Background Information

The Secure Copy Protocol (SCP) is the protocol used for communication from switch processors (SPs) to non−Distributed Forwarding Card (non−DFC) line cards through the Ethernet out of band channel (EOBC) on the Catalyst 6500. SCP or keep−alive polling failures might represent communication issues between the supervisor and the line card.

Whenever a module is powered down, perform these checks:

- Review the logs in order to determine if the module was powered down due to an 'SCP dnld' failure.

- Troubleshoot communication between the supervisor and the line card in question.

Review Logs

Check the logs in order to see if an 'SCP dnld' or keep−alive polling failure is the reason that the module is powered down:

```
%C6KPWR−SP−4−DISABLED: power to module in slot 2 set off (Module Failed SCP dnld)
%C6KPWR−SP−4−DISABLED: power to module in slot 2 set off (Module not responding to
   Keep Alive polling)
```

Troubleshoot Communication

This procedure describes how to troubleshoot the communication between the supervisor and the line card.

1. Check the global SCP counters from the SP−side for any incrementing errors.

```
6500#remote command switch show scp counters
6500−sp#
received packets            = 586786
transmitted packets         = 584442
retransmitted packets       = 13        (increasing re−transmissions indicate
                                         congested EOBC)
loop back packets           = 0
transmit failures           = 0         (increasing transmit failures indicate
                                         congested/stuck EOBC)
recv pkts not for me        = 0
recv pkts to dead process   = 0
recv pkts not enqueuable    = 0         (increasing counters indicate lack of
                                         EOBC buffers)
response has wrong opcode   = 0
response has wrong seqnum   = 0
response is not an ack      = 0
response is too big         = 0
```

2. Check the per−module SCP receive/transmit counters, and check for incrementing SCP retries.

```
6500#remote command switch show scp status
6500−sp#
Rx 586786 ,  Tx 584442 ,  Sap 15
Id     Channel name    current/peak/retry/total  time(queue/process)
−− ------------------ ------------------------  ------------------
0  SCP async: LCP#8     0/ 11/  1/   13            4/   4
1  SCP async: LCP#4     0/ 13/  0/  550           92/ 108
2  SCP async: LCP#2     0/ 34/  0/ 1540          628/ 456
3  SCP async: LCP#5     0/ 17/  1/  716         2228/1252
4  SCP async: LCP#1     0/ 29/  0/  137          200/ 452
5  SCP async: LCP#9     0/ 13/  0/  895          176/ 428
```

3. Check SCP pings from the supervisor to the module in question.

```
6500#remote command switch test scp ping 3
6500−sp#
pinging addr 5(0x5)
assigned sap 0x11
addr 5(0x5) is alive       (Communication between the supervisor and line
                            card is fine)

6500#remote command switch test scp ping 2
```

```
6500−sp#
pinging addr 11(0xB)
assigned sap 0x11
no response from addr 11(0xB)  (Communication between the supervisor
                                and linecard is broken)
```

4. Configure online diagnostics on the line card.

```
6500(config)#diagnostic level complete      (12.1(8a)EX or above)
```

5. Reseat the line card, and review the test results in order to see if any tests failed.

```
6500#show diagnostic result module 2
Current Online Diagnostic Level = Complete
Online Diagnostic Result for Module 2 : PASS
Online Diagnostic Level when Module 2 came up = Complete
```

6. Optional: Use debug commands in order to inspect SCP download events. These debugs may be run to check the SCP download events as a line card comes online. This is an example of a module that is working correctly.

```
6500#remote login switch
6500−sp#debug scp download module 2
6500−sp#show debug
<snip>
SCP download debugging for slot 2 is on
      start_timer_online_action: Start OIR online timer for slot: 2,
         time: 1380 sec
     scp_dnld_module 2 : 0 : 0: during state enabled, got event 5(registered)
 @@@ scp_dnld_module 2 : 0 : 0: enabled −> wait_til_boot_ready
 Stop timer
 Start BOOT_RDY timer for 2 with 30000 msec
     scp_dnld_module 2 : 0 : 0: during state wait_til_boot_ready, got event
        6(boot_ready)
 @@@ scp_dnld_module 2 : 0 : 0: wait_til_boot_ready −> wait_til_downloaded
 Stop timer
 Start DNLD timer for 2 with 120 sec
 (scp_start_download) 2/0
 (scp_start_download) 2/0: Started D/L Process, pid 512
 get_card_image: slot/proc 2/0: UBIN patch image on flash opened
    (microcode:/LCP_CPGBIT)
 No download needed for card at slot 2

     scp_dnld_module 2 : 0 : 0: during state wait_til_downloaded, got event
        4(dnld_completed)
 @@@ scp_dnld_module 2 : 0 : 0: wait_til_downloaded −> wait_til_ready
 Stop timer
 Start EXEC_CODE timer for 2 with 90 sec
 Received Run−ready from slot 2
 scp_download_process_teardown() mypid 512, slot/proc 2/0, image_fd −1
     scp_dnld_module 2 : 0 : 0: during state wait_til_ready, got event
        8(ready)
 @@@ scp_dnld_module 2 : 0 : 0: wait_til_ready −> wait_til_running
 Stop timer
 Start RUN_RDY timer for 5 with 90 sec
     scp_dnld_module 2 : 0 : 0: during state wait_til_running, got
        event 9(running)
 @@@ scp_dnld_module 2 : 0 : 0: wait_til_running −> wait_til_online
 Stop timer
<snip>
```