How to use FCS and CRC Troubleshooting Scripts for ACI

Contents

[Introduction](#)
[Prerequisites To run the Script Manually](#)
[Prerequisites to run the Script from Container](#)
[Steps to execute the scripts](#)

Introduction

ACI follows Cut-Through Switching which means the Packet is already forwarded before the CRC can be computed. These packets are typically stomped and forwarded out as output errors. Because ACI does not drop these packets, the same packet traverses the packet and the stomp CRC counters are incremented on the path. This does not mean all the interfaces that see the CRC are faulty. Therefore, proper triage is needed to isolate the problematic Port/SFP/Fiber. The triage process in now automated through Python scripts resulting in easier troubleshooting and avoiding manual tasks. The scope of this document is to explain how to use the automation scripts are to be used (see attached).

Prerequisites To run the Script Manually

The client machine where the script will be execute from, needs to meet following requirements

a. Python3 should be installed

b. Network access to ACI Domain

c. ACI_CRC_requirements.txt (attached) to be installed. This file is located here.

Download the file (ACI_CRC_requirements.txt) to client machine

Open Terminal and run the command- pip3 install -r ACI_CRC_requirements.txt

```
ABCD-M-G24X:downloads abcd$ pip3 install -r ACI_CRC_requirements.txt

Collecting bcrypt==3.2.0 (from -r ACI_CRC_requirements.txt (line 1))
Downloading
https://files.pythonhosted.org/packages/bf/6a/0afb1e04aebd4c3ceae630a87a55fbfbbd94dea4eaf01e53d3
6743c85f02/bcrypt-3.2.0-cp36-abi3-macosx_10_9_x86_64.whl
Collecting cffi==1.14.6 (from -r ACI_CRC_requirements.txt (line 2))
Downloading
https://files.pythonhosted.org/packages/ca/e1/015e2ae23230d9de8597e9ad8c0b81d5ac181f08f2e6e75774
b7f5301677/cffi-1.14.6-cp38-cp38-macosx_10_9_x86_64.whl (176kB)
|| 184kB 1.4MB/s

**snip**

Successfully installed DateTime-4.3 Pillow-8.3.2 bcrypt-3.2.0 cffi-1.14.6 cryptography-3.4.8
cycler-0.10.0 kiwisolver-1.3.2
matplotlib-3.4.3 numpy-1.21.2 pandas-1.3.2 paramiko-2.7.2 pyparsing-2.4.7 python-dateutil-2.8.2
```

```
pytz-2021.1 six-1.16.0
stdiomask-0.0.5 tabulate-0.8.9 termcolor-1.1.0 zope.interface-5.4.0
```

Prerequisites to run the Script from Container

A container is prepared with the above Python Packages pre-installed.

```
docker login docker.io docker pull aci-stomper docker run -d --name <somename> -p <someport>:80
aci-stomper (on your browser http://ContainerIP:someport) docker ps docker exec -it <Container
ID from previous command> /bin/bash root@6df99d5dbbad:/# cd /home/scripts/
root@6df99d5dbbad:/home/scripts# ls ACI_CRC_Parser.py ACI_CRC_Poller.py
```

Steps to execute the scripts

Please note, there are total two python scripts (ACI_CRC_Poller.py and ACI_CRC_Parser.py). These scripts are available to be downloaded from Cisco DevNet Code Exchange using the below URL.

https://developer.cisco.com/codeexchange/github/repo/CiscoDevNet/ACI-CRC-FCS-Checker

Download both of them on machine/computer, where you want to execute that scripts from.

In this document script-1 refers to ACI_CRC_Poller.py and script-2 refers to ACI_CRC_Parser.py.

1. ACI_CRC_Poller.py will collect CRC and FCS error data in files every five minutes for maximum upto seven days of duration.

Run script-1 (ACI_CRC_Poller.py) from Terminal. Please input the OOB IP address for one of the APICs in given cluster and it's credentials.

ABCD-M-G24X:downloads abcd$ python3 ACI_CRC_Poller.py
Enter the IP address or DNS Name of APIC: 10.197.204.184

Enter the username: admin

Enter the password: ********
Trying to connect to APIC
Connection established to the APIC**

---

2. Script-1 asks for the localtion/path on local machine/computer, where it will store the records. Please enter a valid path, else the script would not run.

Please enter the folder where files have to be stored

**VALID folder format:
EXAMPLE:
Windows-> C:\Users\Admin\Desktop\ACI\
MAC -> /User/admin/Desktop/ACI/**

PLEASE NOTE that data collection and script execution might get impacted if folder format is not as below

---

Enter the absolute path of the folder where the files have to be stored:/Users/abcd/Downloads/FCS_Checker/
<<<<<<<<<<<<<<

3. The script will now ask for end time of execution.

Please enter the time in format **yyyy-mm-dd hh:mm** (per local timezone of fabric), minimum 5 minutes and maximum upto 7 days.

At this time, script-1 starts collecting FCS/CRC errors from the fabric every five minutes (until the endtime specified earlier by the user) and saves data to files at the path specified in earlier input.

```
------------------------------------------------------------------
Enter the End Time until which the script runs(in the format of yyyy-mm-dd hh:mm, current time:2021-09-27 11:27....  maximum upto 2021-10-04 11:27): 2021-09-27 11:32 <<<<<

_____________________________________________________________

The script is executing .....................
The script is executing .....................
ABCD-M-G24X:downloads abcd$
```

4. Upon succesful execution of the first script, it will store raw data files in the location, specified by user in step-2.

Verify the same as shown in below example.

```
ABCD-M-G24X:FCS_Checker kbosu$ pwd
/Users/abcd/Downloads/FCS_Checker

ls -l
total 16
-rw-r--r--@ 1 kbosu  staff  1419 Sep 27 11:28 CRC_FCS_20210927_1128.txt
-rw-r--r--@ 1 kbosu  staff  1419 Sep 27 11:33 CRC_FCS_20210927_1133.txt
ABCD-M-G24X:FCS_Checker abcd$
```

5. Now it's time to execute the second script (ACI_CRC_Parser.py) .

Script-2 is going to use those files created by script-1 and work further.

Please enter the OOB IP address for one of the APICs in given cluster and it's credentials.

Also, enter the same file location, that you entered in step-2 while executing the first script.

```
ABCD-M-G24X:downloads abcd$ python3 ACI_CRC_Parser.py

Enter the IP address or DNS Name of APIC: 10.197.204.184
_____________________________________________________________
Enter the username: admin

_____________________________________________________________
Enter the password: **********
Trying to connect to APIC
Connection established to the APIC

_____________________________________________________________
Please enter the folder where files are stored
Please make sure we have at least two files exists in the directory where you have saved data

_____________________________________________________________
-------------------------------------------------------------------------------------------------------
Enter the absolute path of the folder where the files are stored:/Users/abcd/Downloads/FCS_Checker/

_____________________________________________________________
You have CRC and FCS for the below date range
1.2021-09-27
Fetching first and last file of the same date 20210927
CRC_FCS_20210927_1128.txt
CRC_FCS_20210927_1133.txt

_____________________________________________________________
The script is executing.....

_____________________________________________________________
The script execution has completed
```

6. Script-2 is going to print the data in a tabular format as shown in below example.

Primarily, it is going to list the node interfaces with non-zero CRC and FCS errors, along with the difference in their CRC/FCS counters, during the time interval specified by user. Using LLDP, the script is also going to determine the neighbor device hooked with given interfaces and most importantly, it is going to indicate which node/interface is the source of errors from fabric standpoint and which node interfaces are just seeing CRCs due to Stomp.

From FCS troubleshooting perspective, the one highlighted in "Red" and marked as "Local" is where further troubleshooting should be focused on.

This is likely the interface(s), where bad/corrupted packets are entering into the fabric from and causing the CRCs to be flooded in fabric.

```
+--------+---------+-------------------+-----------+-----------+--------------+--------------+--------------+--------------+------------------------------------
---------------------------------------------+--------------+
| POD_ID | NODE_ID |     NODE_NAME     | NODE_ROLE | INTERFACE | 20210927_1128 | 20210927_1133 | 20210927_1128 | 20210927_1133 |                  NEIGHBOR                  | ERROR SOURCE |
+--------+---------+-------------------+-----------+-----------+-----CRC-------+----CRC Diff---+----FCS--------+---FCS Diff----+---------------------
------------------------------------------------------------+--------------+
|   1    |   302   | bgl-aci06-t2-leaf2 |   leaf    |  eth1/44  | 5002806823759 |   127841888   | 5002806823759 |   127841888   | No LLDP /CDP neighbours found please check physically where this interface connects |    Local     |
|   1    |   101   | bgl-aci06-spine1 |   spine   |  eth1/1   | 2981200154 |  132103050  |      0      |      0      | System:bgl-aci06-t1-leaf1.cisco.com,Interface:Eth1/49 |    Stomp     |
|   1    |   101   | bgl-aci06-spine1 |   spine   |  eth1/2   |    968286    |      0      |      0      |      0      |                  |   Historic   |
|   1    |   201   | bgl-aci06-t1-leaf1 |   leaf    |  eth1/1   |      12      |      0      |      0      |      0      |                  |   Historic   |
|   1    |   201   | bgl-aci06-t1-leaf1 |   leaf    |  eth1/51  | 4999243774529 |      0      |      0      |      0      |                  |   Historic   |
|   1    |   201   | bgl-aci06-t1-leaf1 |   leaf    |  eth1/52  | 5002807353809 |  127841212  |      0      |      0      | System:bgl-aci06-t2-leaf2.cisco.com,Interface:Eth1/49 |    Stomp     |
|   1    |   202   | bgl-aci06-t1-leaf2 |   leaf    |  eth1/51  |    968286    |      0      |      0      |      0      |                  |   Historic   |
|   1    |   301   | bgl-aci06-t2-leaf1 |   leaf    |  eth1/44  | 4999245287405 |      0      | 4999245287405 |      0      |                  |   Historic   |
|   1    |   301   | bgl-aci06-t2-leaf1 |   leaf    |  eth1/49  | 4999823953891 |      0      |      0      |      0      |                  |   Historic   |
|   1    |   302   | bgl-aci06-t2-leaf2 |   leaf    |  eth1/49  | 4999243774529 |      0      |      0      |      0      |                  |   Historic   |
+--------+---------+-------------------+-----------+-----------+--------------+--------------+--------------+--------------+------------------------------------
---------------------------------------------+--------------+
```

7. Additionally, the script is going to provide following options to the users to sort and view granular data, what was collected by script-1 and 2.

User may choose an option between number 1-3 as an input. See the example below.

```
1.Sort the data further
2.View the granular data of an interface
3.Exit

Input the number:
```

In below example, we are going for option 2 which helps us to view granular data for any given interface.

The script will prompt user to enter the respective POD number , Node ID and interface ID from the table printed above (step 6).

Here in this example, we are using 1-302-eth1/44, where POD ID is 1, Node ID is 302 and Interface ID eth1/44. This is the interface

where local FCS was reported by the script, as shown in step-6.

Input the number:2

Enter an interface for which you need granular data(POD_ID-NODE_ID-INTERFACE Example:1-101-eth1/5): 1-302-eth1/44

You have CRC and FCS data in the below date range
1.2021-09-27

Enter the date for which you need granular data(any number from the above list range(1-1)):
In our example, we collected the data only for few minutes of a day, hence we see just one option for dated 27th Sep.

Thus, our input will be "1".

Enter the date for which you need granular data(any number from the above list range(1-1)): 1

+-------+---------------+---------------+
| Time | CRC | FCS |
+-------+---------------+---------------+
| 11:28 | 5002806823759 | 5002806823759 |
| 11:33 | 5002934665647 | 5002934665647 |
+-------+---------------+---------------+

Do you want to continue viewing the granular data(0/1), 1-yes, 0-no:0

Please select any number below to sort the data further or to view granular data of an interface

1.Sort the data further
2.View the granular data of an interface
3.Exit

Input the number:3
ABCD-M-G24X:downloads abcd$