# Utilisation des scripts de dépannage FCS et CRC pour l'ACI

### Table des matières

Introduction

Conditions préalables Pour exécuter le script manuellement

Conditions préalables à l'exécution du script à partir du conteneur

Étapes d'exécution des scripts

## Introduction

L'ACI suit la commutation cut-through, ce qui signifie que le paquet est déjà transféré avant que le CRC puisse être calculé. Ces paquets sont généralement piétinés et transférés en tant qu'erreurs de sortie. Étant donné que l'ACI ne supprime pas ces paquets, le même paquet traverse le paquet et les compteurs de CRC Stop sont incrémentés sur le chemin. Cela ne signifie pas que toutes les interfaces qui voient le CRC sont défectueuses. Par conséquent, un triage approprié est nécessaire pour isoler le port/SFP/fibre problématique. Le processus de triage est désormais automatisé via des scripts Python, ce qui facilite le dépannage et évite les tâches manuelles. L'objectif de ce document est d'expliquer comment utiliser les scripts d'automatisation (voir cijoint).

# Conditions préalables Pour exécuter le script manuellement

La machine cliente à partir de laquelle le script sera exécuté doit répondre aux exigences suivantes

- a. Python3 doit être installé
- b. Accès réseau au domaine ACI
- c. ACI CRC requirements.txt (joint) à installer. Ce fichier se trouve ici.

Téléchargez le fichier (ACI\_CRC\_requirements.txt) sur l'ordinateur client

Ouvrez Terminal et exécutez la commande- pip3 install -r ACI\_CRC\_requirements.txt

<#root>

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-202

Conditions préalables à l'exécution du script à partir du conteneur

Un conteneur est préparé avec les Packages Python ci-dessus préinstallés.

stdiomask-0.0.5 tabulate-0.8.9 termcolor-1.1.0 zope.interface-5.4.0

```
docker login docker.io
docker pull aci-stomper
docker run -d --name

-p

:80 aci-stomper (on your browser http://ContainerIP:someport)
docker ps
docker exec -it

/bin/bash
root@6df99d5dbbad:/# cd /home/scripts/
root@6df99d5dbbad:/home/scripts# ls
ACI_CRC_Parser.py ACI_CRC_Poller.py
```

# Étapes d'exécution des scripts

Veuillez noter qu'il y a au total deux scripts python (ACI\_CRC\_Poller.py et ACI\_CRC\_Parser.py). Ces scripts peuvent être téléchargés à partir de Cisco DevNet Code Exchange à l'aide de l'URL ci-dessous.

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

Téléchargez les deux sur la machine/l'ordinateur à partir duquel vous souhaitez exécuter ces scripts.

Dans ce document, script-1 fait référence à ACI\_CRC\_Poller.py et script-2 fait référence à ACI\_CRC\_Parser.py.

1. ACI\_CRC\_Poller.py recueillera les données d'erreur CRC et FCS dans des fichiers toutes les cinq minutes pour une durée maximale de sept jours.

Exécutez script-1 (ACI\_CRC\_Poller.py) à partir du terminal. Entrez l'adresse IP OOB de l'un des APIC dans le cluster donné et ses informations d'identification.

<#root>

```
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. Le script 1 demande la localisation/le chemin sur l'ordinateur local, où il stockera les enregistrements. Entrez un chemin d'accès valide, sinon le script ne s'exécutera pas.

<#root>

<<<<<<<

```
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. Le script demande maintenant l'heure de fin de l'exécution.

Veuillez saisir l'heure au format yyy-mm-dd hh:mm (par fuseau horaire local du fabric), minimum 5 minutes et maximum jusqu'à 7 jours.

À ce stade, script-1 commence à collecter les erreurs FCS/CRC du fabric toutes les cinq minutes (jusqu'à l'heure de fin spécifiée précédemment par l'utilisateur) et enregistre les données dans des fichiers au niveau du chemin spécifié dans l'entrée précédente.

4. Une fois le premier script exécuté, il stocke les fichiers de données brutes à l'emplacement spécifié par l'utilisateur à l'étape 2.

Vérifiez la même chose que dans l'exemple ci-dessous.

```
<#root>
```

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

ls -1
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. Maintenant, il est temps d'exécuter le deuxième script (ACI\_CRC\_Parser.py).

Script-2 va utiliser les fichiers créés par script-1 et aller plus loin.

Entrez l'adresse IP OOB de l'un des APIC dans le cluster donné et ses informations d'identification.

Entrez également le même emplacement de fichier que celui que vous avez entré à l'étape 2 lors de l'exécution du premier script.

```
<#root>
```

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\_Check

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.....

6. Script-2 va imprimer les données sous forme de tableau, comme illustré ci-dessous.

Il va principalement répertorier les interfaces de noeud avec des erreurs CRC et FCS non nulles, ainsi que la différence dans leurs compteurs CRC/FCS, pendant l'intervalle de temps spécifié par l'utilisateur. Avec le protocole LLDP, le script va également déterminer le périphérique voisin connecté avec des interfaces données et, plus important encore, il va indiquer quel noeud/interface est la source d'erreurs du point de vue du fabric et quelles interfaces de noeud voient juste des CRC en raison de Stomp.

Du point de vue du dépannage FCS, celui mis en évidence en « Rouge » et marqué comme « Local » est celui sur lequel le dépannage doit se concentrer.

Il s'agit probablement de l'interface ou des interfaces à partir desquelles des paquets défectueux/endommagés pénètrent dans le fabric et provoquent l'inondation des CRC dans le fabric.

<#root>

The script execution has completed

	POD_ID	NODE_ID	NODE_NAME	NODE_ROLE	INTERFACE	20210927_1128	20210927_1133	202109
	1	302	bgl-aci06-t2-leaf2	leaf	eth1/44	+CRC   5002806823759	127841888	500280
ı	1	101	bgl-aci06-spine1	spine	eth1/1	2981200154	132103050	
	1	101	bgl-aci06-spine1	spine	eth1/2	968286	0	
Ī	1	201	bgl-aci06-t1-leaf1	leaf	eth1/1	12	0	
Ī	1	201	bgl-aci06-t1-leaf1	leaf	eth1/51	4999243774529	0	
ĺ	1	201	bgl-aci06-t1-leaf1	leaf	eth1/52	5002807353809	127841212	

1	202	bgl-aci06-t1-leaf2	leaf	eth1/	51   968286	0	
1	301	bgl-aci06-t2-leaf1	leaf	eth1/	44   4999245287	405   0	499924
1	301	bgl-aci06-t2-leaf1	leaf	eth1/	49   4999823953	891   0	
1	302	bgl-aci06-t2-leaf2	leaf	eth1/	49   4999243774	529   0	1

7. En outre, le script va fournir les options suivantes aux utilisateurs pour trier et afficher les données granulaires, ce qui a été collecté par script-1 et 2.

L'utilisateur peut choisir une option comprise entre 1 et 3 comme entrée. Voir l'exemple ci-dessous.

<#root>

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

Input the number:

Dans l'exemple ci-dessous, nous allons vers l'option 2 qui nous aide à afficher des données granulaires pour une interface donnée.

Le script invite l'utilisateur à saisir le numéro POD, l'ID de noeud et l'ID d'interface respectifs dans le tableau imprimé ci-dessus (étape 6).

Dans cet exemple, nous utilisons 1-302-eth1/44, où POD ID est 1, Node ID est 302 et Interface ID eth1/44. Il s'agit de l'interface

où la séquence de contrôle de trame locale a été signalée par le script, comme indiqué à l'étape 6.

<#root>

Input the number:2

-----

Enter an interface for which you need granular data(POD\_ID-NODE\_ID-INTERFACE Example:1-101-eth1/5): 1-3

<#root>

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)):

Dans notre exemple, nous avons recueilli les données seulement pour quelques minutes d'une journée, donc nous voyons juste une option pour daté 27 sept.

Par conséquent, notre entrée sera « 1 ».

#### <#root>

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

Time	CRC	FCS
	5002806823759 5002934665647	5002806823759 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\$

#### À propos de cette traduction

Cisco a traduit ce document en traduction automatisée vérifiée par une personne dans le cadre d'un service mondial permettant à nos utilisateurs d'obtenir le contenu d'assistance dans leur propre langue.

Il convient cependant de noter que même la meilleure traduction automatisée ne sera pas aussi précise que celle fournie par un traducteur professionnel.