Comment utiliser les scripts de dépannage FCS et CRC pour ACI

Contenu

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 estompés et transférés en tant qu'erreurs de sortie. Comme l'ACI n'abandonne pas ces paquets, le même paquet traverse le paquet et les compteurs CRC stomp 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 par le biais de scripts Python, ce qui facilite le dépannage et évite les tâches manuelles. Ce document explique comment utiliser les scripts d'automatisation (voir ci-joint).

Conditions requises pour exécuter manuellement le script

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.

Télécharger le fichier (ACI_CRC_Requirements.txt) sur la machine cliente

Ouvrez Terminal et exécutez la commande- 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

Conditions requises pour exécuter le script à partir du conteneur

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

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

Étapes d'exécution des scripts

Veuillez noter qu'il existe 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 l'ordinateur ou l'ordinateur à partir duquel vous voulez exécuter ces scripts.

Dans ce document, le script-1 fait référence à ACI_CRC_Poller.py et le script-2 fait référence à ACI_CRC_Parser.py.

1. ACI_CRC_Poller.py collecte les données d'erreur CRC et FCS dans des fichiers toutes les cinq minutes pendant une durée maximale de sept jours.

Exécutez le script-1 (ACI_CRC_Poller.py) à partir du terminal. Saisissez l'adresse IP OOB de l'un des APIC du cluster donné et ses informations d'identification.

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 demande l'emplacement/le chemin sur l'ordinateur/l'ordinateur local, où il stockera les enregistrements. Entrez un chemin valide, sinon le script ne s'exécutera pas.

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 va maintenant demander l'heure de fin de l'exécution.

Veuillez entrer l'heure au format **aaaa-mm-jj hh:mm** (par fuseau horaire local du tissu), minimum 5 minutes et maximum 7 jours.

À ce stade, le 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 les fichiers au chemin spécifié dans l'entrée précédente.

```
------------------------------------------------------------------
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. 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.

```
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. Il est maintenant temps d'exécuter le deuxième script (ACI_CRC_Parser.py).

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

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

Saisissez également le même emplacement de fichier que celui que vous avez entré à l'étape 2 lors de l'exécution du premier 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 va imprimer les données dans un format tabulaire 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, au cours de l'intervalle de temps spécifié par l'utilisateur. En utilisant LLDP, le script va également déterminer le périphérique voisin connecté à des interfaces données et, plus important encore, il va indiquer quel noeud/interface est la source des erreurs du point de vue du fabric et quelles interfaces de noeud ne voient que des CRC en raison de Stomp.

Du point de vue du dépannage FCS, celui mis en évidence en rouge et marqué en local est celui sur lequel le dépannage doit être axé.

Il s'agit probablement de la ou des interfaces, où des paquets défectueux ou endommagés pénètrent dans le fabric à partir de la structure et provoquent l'inondation des CRC dans le 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. En outre, le script va fournir les options suivantes aux utilisateurs pour trier et afficher des données granulaires, ce qui a été collecté par les scripts 1 et 2.

L'utilisateur peut choisir une option entre le nombre 1 et le nombre 3 comme entrée. Reportez-vous à l'exemple ci-dessous.

```
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 visualiser des données granulaires pour n'importe quelle interface donnée.

Le script invite l'utilisateur à saisir le numéro de 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 ID POD est 1, ID de noeud 302 et ID d'interface 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.

```
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)):
```

Dans notre exemple, nous avons recueilli les données seulement quelques minutes par jour, donc nous ne voyons qu'une seule option pour le 27 septembre.

Ainsi, notre contribution sera « 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$
```