Verwendung von Skripten zur Fehlerbehebung bei FCS und CRC für die ACI

Inhalt

Einleitung

ACI folgt Cut-Through Switching, d. h. das Paket wird bereits weitergeleitet, bevor der CRC berechnet werden kann. Diese Pakete werden in der Regel gestampft und als Ausgabefehler weitergeleitet. Da diese Pakete von der ACI nicht verworfen werden, durchläuft dasselbe Paket das Paket, und die Stomp-CRC-Zähler werden auf dem Pfad inkrementiert. Dies bedeutet nicht, dass alle Schnittstellen, die das CRC sehen, fehlerhaft sind. Daher ist eine geeignete Triage erforderlich, um den problematischen Port/SFP/Fiber zu isolieren. Der Triage-Prozess wurde nun mithilfe von Python-Skripten automatisiert, was die Fehlerbehebung vereinfacht und manuelle Aufgaben vermeidet. Im Rahmen dieses Dokuments wird erläutert, wie die zu verwendenden Automatisierungsskripts verwendet werden sollen (siehe Anhang).

Voraussetzungen Manuelles Ausführen des Skripts

Der Client-Computer, von dem aus das Skript ausgeführt wird, muss folgende Anforderungen erfüllen:

antwort: Python3 sollte installiert werden

b. Netzwerkzugriff auf die ACI-Domäne

c. ACI_CRC_requirements.txt (angehängt) zur Installation. Diese Datei befindet sich hier.

Laden Sie die Datei (ACI_CRC_requirements.txt) auf den Client-Computer herunter.

Öffnen Sie Terminal, und führen Sie den Befehl pip3 install -r ACI_CRC_requirements.txt aus.

<#root>

```
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/0afb1e04aebd4c3ceae630a87a55fbfbbd94dea4eaf01e
Collecting cffi==1.14.6 (from -r ACI_CRC_requirements.txt (line 2))
Downloading https://files.pythonhosted.org/packages/ca/e1/015e2ae23230d9de8597e9ad8c0b81d5ac181f08f2e6e7
|████████████████████████████████| 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
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
stdiomask-0.0.5 tabulate-0.8.9 termcolor-1.1.0 zope.interface-5.4.0
```

Voraussetzungen zum Ausführen des Skripts aus dem Container

Ein Container wird mit den oben vorinstallierten Python-Paketen vorbereitet.

```
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
```

Schritte zur Ausführung der Skripte

Bitte beachten Sie, dass es insgesamt zwei Python-Skripte gibt (ACI_CRC_Poller.py und ACI_CRC_Parser.py). Diese Skripts können über die folgende URL von Cisco DevNet Code Exchange heruntergeladen werden:

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

Laden Sie beide auf den Computer herunter, von dem aus Sie die Skripte ausführen möchten.

In diesem Dokument bezieht sich script-1 auf ACI_CRC_Poller.py und script-2 auf ACI_CRC_Parser.py.

1. ACI_CRC_Poller.py sammelt alle fünf Minuten CRC- und FCS-Fehlerdaten in Dateien für eine Dauer von maximal sieben Tagen.

Führen Sie script-1 (ACI_CRC_Poller.py) von Terminal aus. Geben Sie die OOB-IP-Adresse für einen der APICs im angegebenen Cluster und dessen Anmeldeinformationen ein.

<#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. Skript-1 fragt nach dem Standort/Pfad auf dem lokalen Rechner/Computer, wo die Datensätze gespeichert werden. Geben Sie einen gültigen Pfad ein, da das Skript andernfalls nicht ausgeführt wird.

<#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 belo
-------------------------------------------------------------------------------------------------------


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

/

 <<<<<<<<<<<<<
```

3. Das Skript fragt nun nach der Endzeit der Ausführung.

Bitte geben Sie die Zeit im Format JJJJ-MM-TT hh:mm (pro lokaler Zeitzone des Gewebes) ein, mindestens 5 Minuten und maximal bis zu 7 Tage.

Zu diesem Zeitpunkt beginnt script-1 alle fünf Minuten mit der Erfassung von FCS/CRC-Fehlern aus der Fabric (bis zu der vom Benutzer zuvor angegebenen Endzeit) und speichert Daten in Dateien unter dem in der früheren Eingabe angegebenen Pfad.

<#root>

```
------------------------------------------------------------
Enter the End Time until which the script runs(in the format of yyyy-mm-dd hh:mm, current time:2
_____________________________________________________________
The script is executing ......................
The script is executing ......................
ABCD-M-G24X:downloads abcd$
```

4. Nach erfolgreicher Ausführung des ersten Skripts werden die Rohdatendateien am Speicherort gespeichert, der vom Benutzer in Schritt 2 festgelegt wurde.

Überprüfen Sie den Zustand wie im folgenden Beispiel gezeigt.

<#root>

```
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. Jetzt ist es an der Zeit, das zweite Skript (ACI_CRC_Parser.py) auszuführen.

Script-2 verwendet die Dateien, die von Script-1 erstellt wurden, und arbeitet weiter.

Geben Sie die OOB-IP-Adresse für einen der APICs im angegebenen Cluster und dessen Anmeldeinformationen ein.

Geben Sie außerdem den gleichen Dateispeicherort ein, den Sie in Schritt 2 während der Ausführung des ersten Skripts eingegeben haben.

<#root>

```
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_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.....
_____________________________________________________________
The script execution has completed
```

6. Skript-2 wird die Daten in einem tabellarischen Format drucken, wie im folgenden Beispiel gezeigt.

Hauptsächlich werden die Node-Schnittstellen mit CRC- und FCS-Fehlern ungleich null sowie die Differenz ihrer CRC-/FCS-Zähler während des vom Benutzer festgelegten Zeitintervalls aufgelistet. Mithilfe von LLDP ermittelt das Skript auch das benachbarte Gerät, das mit bestimmten Schnittstellen verbunden ist. Vor allem wird angegeben, welcher Knoten/welche Schnittstelle vom Fabric-Standpunkt aus die Fehlerquelle ist und welche Knotenschnittstellen aufgrund von Stomp nur CRCs sehen.

Was die FCS-Fehlerbehebung angeht, sollte der Bereich, der rot markiert und mit "Lokal" gekennzeichnet ist, für die weitere Fehlerbehebung im Mittelpunkt stehen.

Dies ist wahrscheinlich die Schnittstelle(n), von der böse/beschädigte Pakete in die Fabric eintreten und die eine Flutung der CRCs in der Fabric verursachen.

<#root>

```
+--------+---------+--------------------+-----------+-----------+---------------+---------------+-------
| POD_ID | NODE_ID |     NODE_NAME      | NODE_ROLE | INTERFACE | 20210927_1128 | 20210927_1133 | 202109
+--------+---------+--------------------+-----------+-----------+-----CRC-------+----CRC Diff---+----FCS
|   1    |   302   | bgl-aci06-t2-leaf2 |   leaf    |  eth1/44  | 5002806823759 |   127841888   | 500280
|   1    |   101   |  bgl-aci06-spine1  |   spine   |  eth1/1   |  2981200154   |   132103050   |
|   1    |   101   |  bgl-aci06-spine1  |   spine   |  eth1/2   |    968286     |       0       |
|   1    |   201   | bgl-aci06-t1-leaf1 |   leaf    |  eth1/1   |      12       |       0       |
|   1    |   201   | bgl-aci06-t1-leaf1 |   leaf    |  eth1/51  | 4999243774529 |       0       |
|   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  | 4999245287405 |       0       | 499924
|   1    |   301   | bgl-aci06-t2-leaf1 |   leaf    |  eth1/49  | 4999823953891 |       0       |
|   1    |   302   | bgl-aci06-t2-leaf2 |   leaf    |  eth1/49  | 4999243774529 |       0       |


+--------+---------+--------------------+-----------+-----------+---------------+---------------+------
```

7. Zusätzlich bietet das Skript den Benutzern folgende Optionen zum Sortieren und Anzeigen detaillierter Daten, die von Skript-1 und 2 gesammelt wurden.

Der Benutzer kann eine Option zwischen 1-3 als Eingabe wählen. Siehe folgendes Beispiel.

<#root>

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

Input the number:
```

Im folgenden Beispiel gehen wir zu Option 2 über, mit der wir detaillierte Daten für eine bestimmte Schnittstelle anzeigen können.

Das Skript fordert den Benutzer zur Eingabe der jeweiligen POD-Nummer, der Knoten-ID und der Schnittstellen-ID aus der oben gedruckten Tabelle auf (Schritt 6).

In diesem Beispiel verwenden wir 1-302-eth1/44, wobei die POD-ID 1, die Knoten-ID 302 und die Schnittstellen-ID eth1/44 ist. Dies ist die Schnittstelle.

wobei die lokale Erstauslieferung vom Skript gemeldet wurde, wie in Schritt 6 gezeigt.

<#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)):
```

In unserem Beispiel haben wir die Daten nur für ein paar Minuten pro Tag gesammelt, daher sehen wir nur eine Option für den 27. September.

Daher lautet unser Input "1".

<#root>

```
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$
```

Informationen zu dieser Übersetzung

Cisco hat dieses Dokument maschinell übersetzen und von einem menschlichen Übersetzer editieren und korrigieren lassen, um unseren Benutzern auf der ganzen Welt Support-Inhalte in ihrer eigenen Sprache zu bieten. Bitte beachten Sie, dass selbst die beste maschinelle Übersetzung nicht so genau ist wie eine von einem professionellen Übersetzer angefertigte. Cisco Systems, Inc. übernimmt keine Haftung für die Richtigkeit dieser Übersetzungen und empfiehlt, immer das englische Originaldokument (siehe bereitgestellter Link) heranzuziehen.