

如何使用FCS和CRC故障排除脚本进行ACI

目录

[简介](#)

[手动运行脚本的必备条件](#)

[从容器运行脚本的必备条件](#)

[执行脚本的步骤](#)

简介

ACI遵循直通交换，这意味着在计算CRC之前，已经转发了数据包。这些数据包通常会因输出错误而被丢弃并转发出去。由于ACI不丢弃这些数据包，因此同一数据包会遍历该数据包，并且该路径上的自定义CRC计数器会递增。这并不表示看到CRC的所有接口都有故障。因此，需要正确分类以隔离有问题的端口/SFP/光纤。现在，通过Python脚本自动执行分类过程，从而更轻松地排除故障并避免手动任务。本文档的范围是说明如何使用自动化脚本（请参阅附件）。

手动运行脚本的必备条件

执行脚本的客户端计算机需要满足以下要求

- a. 应安装Python3
- b. 对ACI域的网络访问
- c. 要安装的ACI_CRC_requirements.txt（已附）。此文件位于[此处](#)。

将文件(ACI_CRC_requirements.txt)下载到客户端

打开终端并运行命令 — 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/0afb1e04aebd4c3ceae630a87a55fbfbbd94dea4eaf01e53d36743c85f02/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/e1/015e2ae23230d9de8597e9ad8c0b81d5ac181f08f2e6e75774b7f5301677/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  
cyclor-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
```

从容器运行脚本的必备条件

4.成功执行第一个脚本后，它将在步骤2中用户指定的位置存储原始数据文件。

验证与以下示例所示相同。

```
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.现在是执行第二个脚本(ACI_CRC_Parser.py)的时候了。

脚本2将使用脚本1创建的文件，并进一步工作。

请输入给定集群中某个APIC的OOB IP地址及其凭证。

此外，输入执行第一个脚本时在步骤2中输入的相同文件位置。

```
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.脚本2将以表格格式打印数据，如下例所示。

主要是，它将列出在用户指定的时间间隔内具有非零CRC和FCS错误的节点接口及其CRC/FCS计数器的差异。使用LLDP时，脚本还将确定与给定接口挂钩的邻居设备，最重要的是，它将从交换矩阵的角度指明哪个节点/接口是错误的源，以及哪些节点接口由于Stomp而只看到CRC。

从FCS故障排除的角度来看，以“红色”突出显示并标记为“本地”的故障排除应重点进行进一步故障排除。

这可能是接口，在该接口中，错误/损坏的数据包从进入交换矩阵，并导致CRC在交换矩阵中泛洪。

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.此外，脚本将为用户提供以下选项来排序和查看由脚本1和2收集的精细数据。

用户可以选择数字1-3之间的选项作为输入。请参阅以下示例。

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

Input the number:

在以下示例中，我们将选择选项2，它帮助我们查看任何给定接口的精细数据。

脚本将提示用户从上面打印的表中输入相应的POD编号、节点ID和接口ID（步骤6）。

在本示例中，我们使用1-302-eth1/44，其中POD ID为1，节点ID为302，接口ID eth1/44。这是接口

其中，脚本报告了本地FCS，如步骤6所示。

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)):

在我们的示例中，我们每天只收集几分钟的数据，因此我们只看到一个选项，日期为9月27日。

因此，我们的输入将为“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\$