

# Come utilizzare gli script di risoluzione dei problemi FCS e CRC per ACI

## Sommario

[Introduzione](#)

[Prerequisiti per l'esecuzione manuale dello script](#)

[Prerequisiti per l'esecuzione dello script dal contenitore](#)

[Passi per l'esecuzione degli script](#)

## Introduzione

ACI segue il Cut-Through Switching, il che significa che il pacchetto è già inoltrato prima che il CRC possa essere calcolato. Questi pacchetti vengono in genere reindirizzati e inoltrati come errori di output. Poiché ACI non scarta questi pacchetti, lo stesso pacchetto attraversa il pacchetto e i contatori CRC dello stomp vengono incrementati sul percorso. Ciò non significa che tutte le interfacce che visualizzano il CRC siano difettose. Pertanto, è necessaria una selezione corretta per isolare la porta/SFP/fibra che presenta un problema. Il processo di valutazione è ora automatizzato tramite script Python, semplificando la risoluzione dei problemi ed evitando operazioni manuali. Scopo di questo documento è spiegare come utilizzare gli script di automazione (vedere allegato).

## Prerequisiti per l'esecuzione manuale dello script

Il computer client da cui verrà eseguito lo script deve soddisfare i seguenti requisiti

- r. È necessario installare Python3
- b. Accesso di rete al dominio ACI
- c. ACI\_CRC\_requirements.txt (collegato) da installare. Il file si trova [qui](#).

Scaricare il file (ACI\_CRC\_requirements.txt) sul computer client

Aprire Terminal ed eseguire il comando: 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  
cyclcr-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`

## Prerequisiti per l'esecuzione dello script dal contenitore

Viene preparato un contenitore con i suddetti pacchetti Python preinstallati.

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

## Passi per l'esecuzione degli script

Sono disponibili due script Python (ACI\_CRC\_Poller.py e ACI\_CRC\_Parser.py). Questi script sono disponibili per il download da Cisco DevNet Code Exchange utilizzando l'URL riportato di seguito.

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

Scaricarli entrambi sul computer, dove si desidera eseguire gli script.

In questo documento, script-1 fa riferimento a ACI\_CRC\_Poller.py e script-2 fa riferimento a ACI\_CRC\_Parser.py.

1. ACI\_CRC\_Poller.py raccoglierà i dati degli errori CRC e FCS in file ogni cinque minuti per un massimo di sette giorni di durata.

Eseguire lo script-1 (ACI\_CRC\_Poller.py) da Terminal. Immettere l'indirizzo IP OOB per uno degli APIC nel cluster specificato e le relative credenziali.

```
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 richiede la posizione/il percorso sul computer locale/computer in cui archiviare i record. Immettere un percorso valido, altrimenti lo script non verrà eseguito.

---

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. Lo script richiederà ora l'ora di fine dell'esecuzione.

Immettere l'ora nel formato **aaaa-mm-gg hh:mm** (per fuso orario locale della struttura), da un minimo di 5 minuti a un massimo di 7 giorni.

A questo punto, lo script-1 inizia a raccogliere gli errori FCS/CRC dall'infrastruttura ogni cinque

minuti (fino all'ora di fine specificata in precedenza dall'utente) e salva i dati nei file nel percorso specificato nell'input precedente.

---

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. Una volta completata l'esecuzione del primo script, i file di dati non elaborati verranno memorizzati nella posizione specificata dall'utente nel passaggio 2.

Verificare quanto mostrato nell'esempio seguente.

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. A questo punto è possibile eseguire il secondo script (ACI\_CRC\_Parser.py).

Script-2 utilizzerà i file creati da script-1 e lavorerà ulteriormente.

Immettere l'indirizzo IP OOB per uno degli APIC nel cluster specificato e le relative credenziali.

Inoltre, immettere lo stesso percorso di file immesso al passaggio 2 durante l'esecuzione del primo 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 stamperà i dati in formato tabulare come illustrato nell'esempio seguente.

In particolare, verranno elencate le interfacce nodo con errori CRC e FCS diversi da zero, insieme alla differenza nei contatori CRC/FCS, durante l'intervallo di tempo specificato dall'utente. Utilizzando LLDP, lo script determinerà anche il dispositivo adiacente collegato con determinate interfacce e, cosa più importante, indicherà quale nodo/interfaccia è la fonte di errori dal punto di vista dell'infrastruttura e quali interfacce del nodo stanno semplicemente visualizzando CRC a causa di Stomp.

Dal punto di vista della risoluzione dei problemi FCS, quello evidenziato in "Rosso" e contrassegnato come "Locale" è il punto su cui si dovrebbe focalizzare la risoluzione dei problemi.

È probabile che si tratti dell'interfaccia o delle interfacce, in cui i pacchetti danneggiati o danneggiati entrano nel fabric causando il flooding dei CRC.

```

+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 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. Inoltre, lo script fornirà le seguenti opzioni agli utenti per ordinare e visualizzare i dati granulari, che sono stati raccolti dallo script-1 e dallo script-2.

L'utente può scegliere un'opzione tra 1 e 3 come input. Vedere l'esempio seguente.

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

Input the number:

Nell'esempio seguente viene illustrata l'opzione 2 che consente di visualizzare dati granulari per ogni interfaccia specificata.

Lo script richiederà all'utente di immettere il numero del POD, l'ID del nodo e l'ID dell'interfaccia dalla tabella stampata in precedenza (passaggio 6).

In questo esempio, viene utilizzato 1-302-eth1/44, dove ID POD è 1, ID nodo è 302 e ID interfaccia eth1/44. Questa è l'interfaccia

dove FCS locale è stato segnalato dallo script, come illustrato al passaggio 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)):

Nell'esempio riportato, i dati sono stati raccolti solo per pochi minuti al giorno, pertanto è disponibile un'unica opzione con data 27 settembre.

Pertanto, il nostro contributo sarà "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\$