ACI向けFCSおよびCRCトラブルシューティン グスクリプトの使用方法

内容

<u>概要</u> <u>スクリプトを手動で実行するための前提条件</u> <u>コンテナからスクリプトを実行するための前提条件</u> スクリプトの実行手順

概要

ACIはカットスルースイッチングに従います。つまり、CRCを計算する前にパケットがすでに転送されています。通常 、これらのパケットはストンプされ、出力エラーとして転送されます。ACIはこれらのパケットをドロップしないため 、同じパケットがパケットを通過し、ストンプCRCカウンタがパス上で増加します。これは、CRCを検出するすべての インターフェイスに障害があることを意味しているわけではありません。したがって、問題のあるポート/SFP/ファイバ を切り分けるには、適切なトリアージが必要です。Pythonスクリプトによってトリアージプロセスが自動化され、トラ ブルシューティングが容易になり、手動タスクが不要になりました。このドキュメントでは、自動化スクリプトの使用 方法を説明します(添付を参照)。

スクリプトを手動で実行するための前提条件

スクリプトの実行元となるクライアントマシンは、次の要件を満たす必要があります

a.Python3がインストールされている必要があります

b.ACIドメインへのネットワークアクセス

c. ACI_CRC_requirements.txt(添付)をインストールします。このファイルはここ<u>にあります。</u>

ファイル(ACI_CRC_requirements.txt)をクライアントマシンにダウンロードします

ターミナルを開き、コマンド – pip3 install -r ACI_CRC_requirements.txtを実行します

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

コンテナからスクリプトを実行するための前提条件

コンテナは、上記のPythonパッケージがプリインストールされた状態で準備されています。

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

スクリプトの実行手順

Pythonスクリプト(ACI_CRC_Poller.pyとACI_CRC_Parser.py)は全部で2つあります。 これらのスクリプトは、次の URLを使用してCisco DevNet Code Exchangeからダウンロードできます。

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

これらのスクリプトの実行元となるマシン/コンピュータに、両方をダウンロードします。

このドキュメントでは、script-1はACI_CRC_Poller.pyを指し、script-2はACI_CRC_Parser.pyを指します。

1. ACI_CRC_Poller.pyは、最長7日間、ファイル内のCRCおよびFCSエラーデータを5分ごとに収集します。

ターミナルからscript-1(ACI_CRC_Poller.py)を実行します。 特定のクラスタ内のAPICの1つのOOB IPアドレスとそのク レデンシャルを入力してください。

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は、ローカルマシン/コンピュータ上のローカル/パスを要求し、レコードを格納します。 有効なパスを入力 してください。入力しないと、スクリプトは実行されません。

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/ <<<<<<<

を選択します。 スクリプトは、実行の終了時間を要求します。

yyyy-mm-dd hh:mm(ファブリックのローカルタイムゾーンごと)の形式で、最小5分、最大7日 間の時間を入力してください。

この時点で、script-1は5分ごとに(ユーザによって指定された終了時間まで)ファブリックから FCS/CRCエラーの収集を開始し、以前の入力で指定されたパスのファイルにデータを保存します 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.最初のスクリプトが正常に実行されると、ステップ2でユーザが指定した場所にrawデータファ イルが保存されます。

次の例と同じことを確認します。

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

Is -I 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.次に、2番目のスクリプト(ACI_CRC_Parser.py)を実行します。

Script-2では、script-1によって作成されたファイルを使用して、さらに作業を進めます。

特定のクラスタ内のいずれかのAPICのOOB IPアドレスとそのクレデンシャルを入力してください。

また、ステップ2で最初のスクリプトの実行中に入力したファイルの場所と同じ場所を入力します。

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では、次の例に示すように、データが表形式で印刷されます。

主に、ユーザが指定した時間間隔の間に、CRCおよびFCSエラーがゼロ以外のノードインターフェイスと、それらのCRC/FCSカウンタの差異がリストされます。LLDPを使用すると、スクリプトは特定のインターフェイスに接続されたネイバーデバイスも判別します。最も重要なのは、ファブリックの観点から、どのノード/インターフェイスがエラーの原因で、どのノードインターフェイスがStempによりCRCを参照しているかを示します。

FCSのトラブルシューティングの観点から見ると、「赤」で強調表示され、「ローカル」とマー クされているものは、さらにトラブルシューティングを行う必要があります。

これは、不良または破損したパケットがからファブリックに入り、ファブリック内でCRCがフラ ッディングされるインターフェイスである可能性があります。

POD_ID NODE_ID NODE_NA	ME NOD	E_ROLE IN	TERFACE	: 2021 0	927_11	28 20	21092	27_113	33 20210927_1128
20210927_1133	NEIGHB	DR		ER	ROR S	OURCE	- 1		
+++++	+	+CRC	+CR	C Diff	+FC	S	+FC	S Diff	f+
	+	+							
1 302 bgl-aci06-t2-leaf2	leaf eth1/	44 500280	6823759	12784´	888	500280	68237	759	127841888 <mark>N</mark> o
LLDP /CDP neighbours found plea	ise check phy	sically whe	re this inte	erface co	onnects	s Lo	cal	1	
1 101 bgl-aci06-spine1	spine eth	1/1 29812	200154	132103	050	0	1	0	
System:bgl-aci06-t1-leaf1.cisco.co	m,Interface:E	th1/49	Sto	omp					
1 101 bgl-aci06-spine1 Historic	spine eth 	1/2 9682	286	0	0	0	I		
1 201 bgl-aci06-t1-leaf1	leaf eth1/	1 12	0	0		0			
Historic									
1 201 bgl-aci06-t1-leaf1	leaf eth1/	51 499924	3774529	0	0	1	0	Ι	
Histor	ic								
1 201 bgl-aci06-t1-leaf1	leaf eth1/	52 500280	7353809	12784 ⁻	212	0		0	1
System:bgl-aci06-t2-leaf2.cisco.co	m,Interface:E	th1/49	Sto	omp					
1 202 bgl-aci06-t1-leaf2	leaf eth1/	51 96828	36 0	ן כ	0	0			
Historic	1								
1 301 bgl-aci06-t2-leaf1	leaf eth1/	44 499924	5287405	0	4999	245287	405	0	
1	Historic								
1 301 bgl-aci06-t2-leaf1	leaf eth1/	49 499982	3953891	0	0		0	1	
Histor	ic								
1 302 bgl-aci06-t2-leaf2	leaf eth1/	49 499924	3774529	0	0		0	1	
Histor	ic	-			-	-		-	

7.さらに、スクリプトは、script-1および2によって収集された詳細なデータをソートおよび表示 するための次のオプションをユーザに提供します。

ユーザは入力として1~3のオプションを選択できます。下の例を参照してください。

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

Input the number:

次の例では、任意のインターフェイスの詳細なデータを表示するのに役立つオプション2を示しま す。

このスクリプトは、上記の表に記載されている各POD番号(Node ID)、およびインターフェイス IDを入力するようユーザに促します(ステップ6)。

この例では、1-302-eth1/44を使用しています。ここで、POD IDは1、ノードIDは302、インター フェイスIDはeth1/44です。これはインターフェイスです

ローカルFCSは、ステップ6で示すように、スクリプトによって報告されました。

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)): この例では、データを1日の数分だけ収集したため、日付が9月27日のオプションが1つだけ表示 されます。

したがって、入力は「1」になります。

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\$