PythonでIOS-XEルーティングプロトコルフラッピングログを収集する

内容

はじめに

このドキュメントでは、プロトコルのフラップ時にOSPF、EIGRP、およびIS-ISのログを収集するようにPythonスクリプトを設定する方法について説明します。

前提条件

要件

次のトピックに精通していることが推奨されます。

- アプリケーションホスティングの設定
- OSPF
- EIGRP
- IS-IS
- VI エディタ

使用するコンポーネント

このドキュメントの情報は、Cisco IOS XEソフトウェアバージョン17に基づくものです。

このドキュメントの情報は、特定のラボ環境にあるデバイスに基づいて作成されたものです。このドキュメントで使用するすべてのデバイスは、クリアな（デフォルト）設定で作業を開始しています。本稼働中のネットワークでは、各コマンドによって起こる可能性がある影響を十分確認してください。

注：このドキュメントでは、該当する詳細は扱いません。詳細については、参照先のリンクを参照してください。

設定

コンフィギュレーション

TACサービスリクエストをオープンする際は、関連情報を収集して時間を節約することが非常に重要です。場合によっては、デバイスから収集できる基本的な出力の中に障害のヒントが含まれていることがあります。このドキュメントでは、Pythonスクリプトを使用してこのデータを取得する方法の例を示します。OSPF、EIGRP、およびIS-ISの3つのプロトコルが考慮されます。

ステップ 1：最初に行う必要があるのは、guestshellを設定して有効にすることです。

```
Router(config)#iox
Router(config)#interface VirtualPortGroup 0
Router(config-if)#ip address 192.0.2.1 255.255.255.252
Router(config-if)#exit
Router(config)#
Router(config)#app-hosting appid guestshell
Router(config-app-hosting)#app-vnic gateway0 virtualportgroup 0 guest-interface 0
Router(config-app-hosting-gateway0)#guest-ipaddress 192.0.2.2 netmask 255.255.255.252
Router(config-app-hosting)#app-default-gateway 192.0.2.1 guest-interface 0
Router(config)#end
```

この設定では、次の3つの重要なステップがあります。

1. IOXサービスを有効にします。ゲストシェルを有効にするには、これが必要です。

2. ゲストシェルデフォルトゲートウェイのデフォルトゲートウェイとして機能するVirtualPortGroupを設定します。

3. ゲストシェルのアプリケーションホスティングを設定します。VirtualPortGroupが機能する場所

は、設定から確認できます。

ステップ 2次に、特権モードからゲストシェルを有効にする必要があります。

```
Router#guestshell enable
Interface will be selected if configured in app-hosting
Please wait for completion
guestshell installed successfully
Current state is: DEPLOYED
guestshell activated successfully
Current state is: ACTIVATED
guestshell started successfully
Current state is: RUNNING
Guestshell enabled successfully

Router#
*Jun 15 21:31:31.499: %IM-6-IOX_INST_INFO: R0/0: ioxman: IOX SERVICE guestshell LOG: Guestshell is up a
```

すべてが正しく設定されている場合は、前の例のログを確認する必要があります。

ステップ 3これで、Pythonスクリプトを設定する準備が整いました。コマンドguestshellを特権モードで実行します。次の例のように、プロンプトが表示されます。

```
Router#guestshell
[guestshell@guestshell ~]$
```

ステップ 4vi工ディタを使用してファイルを作成し、有効にしたプロトコルに基づいてスクリプトを設定します。

```
[guestshell@guestshell ~]$ vi ospf.py
```

このウィンドウが表示されます

```
~
~
~
~
~
~
~
"ospf.py" 0L, 0C
```

ステップ 5テキストを挿入するには「i」を押します。スクリプトを貼り付け、escキーを押してから、:wqという文字を入力します。

```
~
from cli import cli
from time import sleep

cli("enable")
cli("debug ip ospf hello")
cli("debug ip ospf adj")
cli("show ip ospf interface | append bootflash:Router-ospf-logs.txt")
cli("show ip ospf neighbor | append bootflash:Router-ospf-logs.txt")
cli("show interfaces | append bootflash:Router-ospf-logs.txt")
cli("show logging | append bootflash:Router-ospf-logs.txt")
cli("show tech | append bootflash:Router-showtech.txt")
sleep(30)
cli("undebug all")
~
~
~
~
"ospf.py" [New] 14L, 458C written
[guestshell@guestshell ~]$
```

exitコマンドを使用してゲストシェルを終了します。

確認

スクリプトをテストします。exitコマンドを使用してゲストシェルを終了します。次に、guestshell run python3 ospf.py

```
F340.20.09-8500-1#guestshell run python3 ospf.py
```

OSPF、EIGRP、IS-ISの3つのプロトコルすべてのスクリプトを次に示します。

OSPF

```
from cli import cli
from time import sleep

cli("enable")
cli("debug ip ospf hello")
cli("debug ip ospf adj")
cli("show ip ospf interface | append bootflash:Router-ospf-logs.txt")
cli("show ip ospf neighbor | append bootflash:Router-ospf-logs.txt")
```

```
cli("show interfaces | append bootflash:Router-ospf-logs.txt")
cli("show logging | append bootflash:Router-ospf-logs.txt")
cli("show tech | append bootflash:Router-showtech.txt")
sleep(30)
cli("undebug all")
```

EIGRP

```
from cli import cli
from time import sleep

cli("enable")
cli("debug eigrp packet")
cli("show ip eigrp neighbor | append bootflash:Router-eigrp-logs.txt")
cli("show ip eigrp interface | append bootflash:Router-eigrp-logs.txt")
cli("show interfaces | append bootflash:Router-eigrp-logs.txt")
cli("show logging | append bootflash:Router-eigrp-logs.txt")
cli("show tech | append bootflash:Router-showtech.txt")
sleep(30)
cli("undebug all")
```

IS-IS

```
from cli import cli
from time import sleep

cli("enable")
cli("debug isis adj-packet")
cli("show isis neighbor detail | append bootflash:Router-isis-logs.txt")
cli("show clns neighbor detail | append bootflash:Router-isis-logs.txt")
cli("show clns interface | append bootflash:Router-isis-logs.txt")
cli("show interfaces | append bootflash:Router-isis-logs.txt")
cli("show logging | append bootflash:Router-isis-logs.txt")
cli("show tech | append bootflash:Router-showtech.txt")
sleep(30)
cli("undebug all")
```

syslogパターンが観察された後でPythonスクリプトを実行するEEMスクリプトを使用して、ログ収集を自動化できます。次のセクションでは、このタスクを実行するためにPythonスクリプトとともに設定できるEEMスクリプトを使用します。

OSPF

```
event manager applet ospf-flap authorization bypass
event syslog pattern "%OSPF-5-ADJCHG:.*from FULL to DOWN" maxrun 120 ratelimit 120
action 010 cli command "enable"
```

```
action 020 cli command "guestshell run python3 ospf.py"
action 030 exit
```

EIGRP

```
event manager applet eirgp-flap authorization bypass
event syslog pattern "%DUAL-5-NBRCHANGE: EIGRP.*Neighbor.*is down" maxrun 120 ratelimit 120
action 010 cli command "enable"
action 020 cli command "guestshell run python3 eigrp.py"
action 030 exit
```

IS-IS

```
event manager applet isis-flap authorization bypass
event syslog pattern "%CLNS-5-ADJCHANGE: ISIS: Adjacency to.*Down" maxrun 120 ratelimit 120
action 010 cli command "enable"
action 020 cli command "guestshell run python3 isis.py"
action 030 exit
```

注：これらのスクリプトで収集されるコマンドは、基本的な初期情報を提供します。TACサービスリクエストをオープンする際、必要に応じて、TACエンジニアから詳細な調査を依頼される場合があります。

参照リンク

- ゲストシェル
- Python API

翻訳について
シスコは世界中のユーザにそれぞれの言語でサポート コンテンツを提供するために、機械と人による翻訳を組み合わせて、本ドキュメントを翻訳しています。ただし、最高度の機械翻訳であっても、専門家による翻訳のような正確性は確保されません。シスコは、これら翻訳の正確性について法的責任を負いません。原典である英語版（リンクからアクセス可能）もあわせて参照することを推奨します。