Verzamel IOS-XE Routing Protocol Flapping Logs met Python

Inhoud

Inleiding

Voorwaarden

- Vereisten
- Gebruikte componenten

Configureren

- Configuraties

Verifiëren

Referentielinks

Inleiding

In dit document wordt beschreven hoe u Python-scripts configureert om OSPF-, EIGRP- en IS-IS-logs te verzamelen wanneer de protocollen flappen.

Voorwaarden

Vereisten

Cisco raadt u aan bekend te zijn met de genoemde onderwerpen:

- App-hostingconfiguratie
- OSPF
- EIGRP
- IS-IS
- VI-editor

Gebruikte componenten

De informatie in dit document is gebaseerd op Cisco IOS XE-softwareversie 17.

De informatie in dit document is gebaseerd op de apparaten in een specifieke laboratoriumomgeving. Alle apparaten die in dit document worden beschreven, hadden een opgeschoonde (standaard)configuratie. Als uw netwerk live is, moet u zorgen dat u de potentiële impact van elke opdracht begrijpt.

Opmerking: Dit document gaat niet in op de apphosting-details. Meer informatie is te vinden in de links waarnaar wordt verwezen.

Configureren

Configuraties

Wanneer u een TAC-zaak opent, is het erg belangrijk om relevante informatie te verzamelen om tijd te besparen. Soms ligt de aanwijzing voor een storing in enkele basisuitgangen die u van het apparaat kunt verzamelen. In dit document vindt u voorbeelden van hoe u Python-scripts kunt gebruiken om deze gegevens te verkrijgen. Er zijn drie protocollen: OSPF, EIGRP en IS-IS.

Stap 1. Het eerste wat u moet doen is het configureren en inschakelen van guestshell.

```
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
```

In deze configuratie zijn er drie belangrijke stappen:

1. IOX-service inschakelen. Dit is nodig om guestshell mogelijk te maken.

2. Configureer de VirtualPortGroup die fungeert als de standaardgateway voor de guestshell-standaardgateway.

3. App-hosting configureren voor de gastenshell. U kunt aan de configuraties zien waar de

VirtualPortGroup in het spel komt.

Stap 2. Vervolgens moet u guestshell inschakelen vanuit de voorkeursmodus.

```
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
```

Als alles goed is geconfigureerd, moet u het logboek in het vorige voorbeeld zien.

Stap 3. Nu ben je klaar om de Python-scripts te configureren. Voer de opdracht guestshell in de voorkeursmodus. U ziet de prompt zoals in het volgende voorbeeld:

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

Stap 4. Maak een bestand met vi-editor en configureer de scripts op basis van de protocollen die u hebt ingeschakeld.

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

Dit venster verschijnt

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

Stap 5. Druk op "i" om tekst in te voegen. Plak het script en druk op "esc" en voer de tekens in: wq

```
~
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 ~]$
```

Sluit de guestshell af met de opdracht exit.

Verifiëren

Test het script. Verlaat de guestshell met het commando exit. Voer vervolgens guestshell uit met python3 ospf.py

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

Hier zijn de scripts voor alle drie de protocollen; OSPF, EIGRP en IS-IS.

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")
```

U kunt de logboekverzameling automatiseren met EEM-scripts die de Python-scripts uitvoeren nadat syslog-patronen zijn waargenomen. In de volgende sectie hebt u de EEM-scripts die u samen met de python-scripts kunt configureren om deze taak uit te voeren.

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
```

Opmerking: De opdrachten die in deze scripts worden verzameld, bevatten basisinformatie voor het begin. Wanneer u een TAC-zaak opent, kan meer informatie worden opgevraagd door TAC-ingenieurs om verder te onderzoeken indien nodig.

Referentielinks

- guestshell
- Python-API

Over deze vertaling

Cisco heeft dit document vertaald via een combinatie van machine- en menselijke technologie om onze gebruikers wereldwijd ondersteuningscontent te bieden in hun eigen taal. Houd er rekening mee dat zelfs de beste machinevertaling niet net zo nauwkeurig is als die van een professionele vertaler. Cisco Systems, Inc. is niet aansprakelijk voor de nauwkeurigheid van deze vertalingen en raadt aan altijd het oorspronkelijke Engelstalige document (link) te raadplegen.