

Automatisieren der Cisco IOS XE- Befehlssequenzen mithilfe von TCL und EEM

Einleitung

In diesem Dokument wird ein Verfahren zum Erstellen eines eindeutigen EEM-Skripts beschrieben, mit dem eine beliebige, vom Netzwerkadministrator definierte Befehlsfolge ausgeführt werden kann.

Voraussetzungen

Anforderungen

Cisco empfiehlt, dass Sie über Kenntnisse in folgenden Bereichen verfügen:

- Kenntnis der Cisco IOS® XE-Prozesse
- Grundlagen von Embedded Event Manager (EEM)-Scripting
- Vertrautheit mit Cisco Catalyst Switches
- Grundkenntnisse von TCL-Scripting

Verwendete Komponenten

Die Informationen in diesem Dokument basieren auf dem Cisco Catalyst Switch der Serie 9600X (CS-C9600X) mit der IOS XE Version 17.18.2.

Die Informationen in diesem Dokument beziehen sich auf Geräte in einer speziell eingerichteten Testumgebung. Alle Geräte, die in diesem Dokument benutzt wurden, begannen mit einer gelöschten (Nichterfüllungs) Konfiguration. Wenn Ihr Netzwerk in Betrieb ist, stellen Sie sicher, dass Sie die möglichen Auswirkungen aller Befehle kennen.

Hintergrundinformationen

Der Administrator erstellt eine Textdatei mit der gewünschten Befehlsfolge. Das EEM-Skript ruft diese Datei dann bei einem angegebenen Trigger auf und führt alle in der Textdatei aufgeführten

Befehle aus.

Konfiguration und schrittweise Vorgehensweise

Schritt 1. Erstellen einer Textdatei mit Befehlsfolge

1. Erstellen Sie eine Textdatei mit den erforderlichen Befehlen in der gewünschten Reihenfolge.
2. Kopieren Sie die Datei in den Flash-Speicher des Geräts.
3. Verwenden Sie FTP-, TFTP- oder TCL-Scripting, um die Datei zu erstellen oder zu übertragen.

Erstellen einer Textdatei mithilfe von TCL-Skripts

Syntax:

```
<#root>
```

```
tclsh  
puts [open "flash:COMMANDS_TO_EXECUTE.txt" w+] {
```

```
Enter Commands to Execute
```

```
}  
exit
```

Konfigurationsbeispiel:

So erstellen Sie eine Sequenz von Befehlen, die die Schnittstelle FiftyGigE1/1/0/3 herunterfährt:

```
C9600-Silicon1-SVL#tclsh  
C9600-Silicon1-SVL(tcl)#puts [open "flash:COMMANDS_TO_EXECUTE.txt" w+] {  
configure terminal  
interface fiftyGigE 1/1/0/3  
shutdown  
end  
show run interface fiftyGigE 1/1/0/3  
}  
C9600-Silicon1-SVL(tcl)#exit  
C9600-Silicon1-SVL#
```

Überprüfen Sie die Datei im Flash-Speicher:

```
C9600-Silicon1-SVL#dir flash:  
Directory of flash:/
```

```
357117 -rw- 101 Jan 25 2026 21:37:09 +05:30 COMMANDS_TO_EXECUTE.txt
```

Inhalt von COMMANDS_TO_EXECUTE.txt

```
configure terminal  
interface fiftyGigE 1/1/0/3  
shutdown  
end  
show run interface fiftyGigE 1/1/0/3
```

Diese Datei kann entweder durch TCL-Scripting erstellt oder über FTP, TFTP oder RCP übertragen werden. Die Datei muss nach Bedarf bearbeitet werden, damit sie eine beliebige Folge von Befehlen enthält, die vom Netzwerkadministrator benötigt werden.

Schritt 2: Ausführen der Datei mithilfe des EEM-Skripts

Erstellung des EEM-Skripts

Dieses EEM-Applet führt alle Befehle aus der angegebenen Datei COMMANDS_TO_EXECUTE aus.

```
event manager applet Cisco_Live authorization bypass  
event none maxrun 1000  
action 010 cli command "enable"  
action 020 cli command "term exec prompt timestamp"  
action 030 cli command "term exec prompt expand"  
action 040 file open fileout1 flash:OUTPUT.txt a+  
action 050 file open filein1 flash:COMMANDS_TO_EXECUTE.txt r  
action 060 file read filein1 input  
action 070 foreach line $input "\n"  
action 080 cli command "$line"  
action 090 file write fileout1 "$_cli_result"  
action 100 end  
action 110 file close fileout1  
action 120 file close filein1
```

Erläuterung des EEM-Skripts

++ Event Manager-Applet Cisco_Live-Autorisierungsumgehung:

Appletname des Event Managers: Registriert ein EEM-Applet mit dem angegebenen Namen ('Cisco_Live' in diesem Fall).

Autorisierungsumgehung: Optionale Schlüsselwörter, die das Applet angeben, müssen die AAA-Autorisierung umgehen.

++ Veranstaltung keine maxrun 1000:

Veranstaltung keine: Dadurch wird angegeben, dass die EEM-Richtlinie bei der Ereigniserkennung "none" registriert wird, d. h., dass die Richtlinie nicht durch ein automatisches Ereignis ausgelöst wird, sondern manuell ausgeführt werden soll.

Maxrun 1000: Dieser optionale Parameter legt die maximale Laufzeit für das Applet oder die Richtlinie auf 1000 Sekunden fest. Wenn nicht angegeben, beträgt die standardmäßige maximale Laufzeit 20 Sekunden.

++ action 020 cli-Befehl term exec prompt timestamp:

Aktion 200: Dies ist die Bezeichnung oder Sequenzkennung für die Aktion im EEM-Skript.

term exec prompt timestamp ist ein CLI-Befehl, der Timestamps für die Eingabeaufforderung in der aktuellen Terminalsitzung aktiviert.

Diese Funktion hilft bei der Fehlerbehebung und Überwachung, indem sie für jeden ausgeführten Befehl eine Zeitreferenz bereitstellt.

Der Zeitstempel wird vor der Ausgabe von Befehlen angezeigt, was die Korrelation von Ereignissen und Befehlsausführungen mit bestimmten Zeiten erleichtert.

++ action 030 cli command term exec prompt expand:

term exec prompt expand aktiviert den erweiterten Aufforderungsmodus für die aktuelle Terminalsitzung.

Außerdem wird das Terminal so konfiguriert, dass die vollständige Eingabeaufforderung ohne Kürzung oder Abkürzung angezeigt wird.

Es hilft bei Skripten oder Fehlerbehebungsszenarien, bei denen der vollständige Aufforderungskontext für die Übersichtlichkeit oder Analyse benötigt wird.

Der Befehl wirkt sich nur auf die aktuelle Terminalsitzung aus und bleibt nach dem Ende der Sitzung nicht bestehen.

++ Aktion 040 Datei öffnen fileout1 flash:OUTPUT.txt a+:

Datei öffnen: Gibt den Vorgang zum Öffnen einer Datei an.

Fileout1: Dies ist der Dateideskriptor oder der Handlename, der im Skript für den Verweis auf die geöffnete Datei verwendet wird.

Flash:OUTPUT.txt: Dies ist der Pfad und der Dateiname im Flash-Dateisystem, in dem sich die Datei befindet oder erstellt wird.

a+: Dies ist der Dateizugriffsmodus, d. h. die Datei wird zum Lesen und Anhängen geöffnet. Wenn die Datei nicht vorhanden ist, wird sie erstellt. In die Datei geschriebene Daten werden am Ende angehängt, ohne den vorhandenen Inhalt zu kürzen.

++ action 050 file open file1 flash:COMMANDS_TO_EXECUTE.txt r:

Datei 1: Dies ist der Dateideskriptor oder der Handlename, der im Skript für den Verweis auf die geöffnete Datei verwendet wird.

Flash:COMMANDS_TO_EXECUTE.txt: Dies ist der Pfad und der Dateiname im Flash-Dateisystem, in dem sich die Datei befindet.

r: Dies ist der Dateizugriffsmodus, d. h. die Datei wird nur zum Lesen geöffnet.

++ Aktion 060 Datei Datei lesen1 Eingabe:

Datei gelesen: Gibt den Vorgang an, der aus einer Datei gelesen werden soll.

Datei 1: Dies ist der Dateideskriptor oder das Handle, der bzw. das beim Öffnen der Datei zugewiesen wurde (z. B. mit Aktion 050 file open file1. Flash:COMMANDS_TO_EXECUTE.txt r).

Eingabe: Dies ist der Variablenname, unter dem die gelesenen Daten zur Verwendung in nachfolgenden Skriptaktionen gespeichert werden.

++ Aktion 070 für jede Zeile \$input '\n':

Foreach: Dieser Befehl gibt eine Iteration über eine Zeichenfolge an.

Anschluss: Dies ist die Iteratorvariable, die jedes Token (jede Zeile) während jeder Iteration enthält.

\$input: Dies ist die Eingabezeichenfolgenvariable, die mehrere Zeilen oder Einträge enthält.

'\n': Dies ist das Trennzeichen, mit dem die Eingabezeichenfolge in Token aufgeteilt wird, in diesem Fall das Zeilenendezeichen.

Jede Zeile ist ein Token und \n ist in diesem Fall der Splitter.

++ Aktion 080 CLI-Befehl \$line:

CLI-Befehl: Gibt an, dass die Aktion die Ausführung eines CLI-Befehls auf dem Gerät ist.

\$Zeile: Die Variable mit der auszuführenden CLI-Befehlszeichenfolge. Die Anführungszeichen stellen sicher, dass die gesamte Zeichenfolge als ein einziger Befehl behandelt wird.

++ action 090 file write fileout1 '\$_cli_result':

Datei schreiben: Der EEM-Befehl zum Schreiben von Daten in eine Datei.

Fileout1: Der Dateihandler, der zuvor durch einen Befehl wie file open fileout1 flash:OUTPUT.txt a+ geöffnet wurde, stellt die Datei dar, in die Daten geschrieben werden.

'\$_cli_result': Die Variable, die die CLI-Befehlsausgabe oder beliebige Zeichenfolgendaten enthält, die in die Datei geschrieben werden sollen. Die Anführungszeichen stellen sicher, dass der gesamte Inhalt als eine einzige Zeichenfolge behandelt wird.

Dieser Befehl wird in der Regel verwendet, nachdem CLI-Befehle ausgeführt und ihre Ausgabe in \$_cli_result erfasst wurden, sodass das Skript die Ausgabe in einer Datei im Flash-Speicher des Geräts protokollieren oder archivieren kann.

++ Aktion 100 Ende:

End: Das Schlüsselwort, das das Ende eines bedingten Blocks angibt, z. B. if, else oder while.

++ Aktion 120 Datei schließen Datei1:

Datei schließen: Der Vorgang zum Schließen der Dateibeschreibung.

Datei 1: Der Bezeichner des zu schließenden Dateideskriptors, der zuvor im Skript geöffnet werden muss.

EEM-Skriptausführung

So lösen Sie das EEM-Skript manuell aus:

```
C9600-Silicon1-SVL#event manager run Cisco_Live
```

In diesem Beispiel wird das EEM-Skript manuell ausgelöst. Das Skript kann einen beliebigen Trigger gemäß den Netzwerkanforderungen enthalten.

Ergebnis der Dateiausführung

In der Ausgabedatei werden die ausgeführten Befehle und ihre Ergebnisse angezeigt:

```
<#root>
```

```
C9600-Silicon1-SVL#more flash:OUTPUT.txt
```

```
Enter configuration commands, one per line. End with CNTL/Z.
```

```
C9600-Silicon1-SVL(config)#
```

```
C9600-Silicon1-SVL(config-if)#
```

```
C9600-Silicon1-SVL(config-if)#
```

```
C9600-Silicon1-SVL#Load for five secs: 0%/0%; one minute: 1%; five minutes: 1%
```

```
No time source, *22:01:35.406 IST Sun Jan 25 2026
```

```
----- show running-config interface FiftyGigE1/1/0/3 -----
```

Building configuration...

Current configuration : 127 bytes

!

```
interface FiftyGigE1/1/0/3
description To-Host3-Ten 1/1/3
switchport access vlan 10
switchport mode access
```

```
shutdown
```

```
end
```

Die Schnittstelle FiftyGigE1/1/0/3 wird gemäß der Befehlssequenz erfolgreich heruntergefahren.

Zusammenfassung

In diesem Dokument wurde gezeigt, wie TCL-Scripting und das EEM von Cisco kombiniert werden können, um ein flexibles, einzelnes EEM-Skript zu erstellen, das jede beliebige Sequenz von Befehlen ausführen kann, die in einer Textdatei gespeichert sind. Dieser Ansatz ermöglicht es Netzwerkadministratoren, Befehlssequenzen extern zu definieren und ihre Ausführung automatisch oder manuell auszulösen. Dadurch werden die Automatisierung und Betriebseffizienz auf Cisco Catalyst Switches mit Cisco IOS XE optimiert.

Informationen zu dieser Übersetzung

Cisco hat dieses Dokument maschinell übersetzen und von einem menschlichen Übersetzer editieren und korrigieren lassen, um unseren Benutzern auf der ganzen Welt Support-Inhalte in ihrer eigenen Sprache zu bieten. Bitte beachten Sie, dass selbst die beste maschinelle Übersetzung nicht so genau ist wie eine von einem professionellen Übersetzer angefertigte. Cisco Systems, Inc. übernimmt keine Haftung für die Richtigkeit dieser Übersetzungen und empfiehlt, immer das englische Originaldokument (siehe bereitgestellter Link) heranzuziehen.