

Automatización de las secuencias de comandos de Cisco IOS XE mediante TCL y EEM

Introducción

Este documento describe un método para crear un script EEM único y único capaz de ejecutar cualquier secuencia de comandos definida por un administrador de red.

Prerequisites

Requirements

Cisco recomienda que tenga conocimiento sobre estos temas:

- Conocimiento de las operaciones de Cisco IOS® XE
- Comprensión del scripting de Embedded Event Manager (EEM)
- Familiaridad con los switches Catalyst de Cisco
- Conocimiento básico de scripts TCL

Componentes Utilizados

La información de este documento se basa en el switch Catalyst de Cisco serie 9600X (CS-C9600X) que ejecuta la versión IOS XE 17.18.2.

La información que contiene este documento se creó a partir de los dispositivos en un ambiente de laboratorio específico. Todos los dispositivos que se utilizan en este documento se pusieron en funcionamiento con una configuración verificada (predeterminada). Si tiene una red en vivo, asegúrese de entender el posible impacto de cualquier comando.

Antecedentes

El administrador crea un archivo de texto sin formato que contiene la secuencia de comandos deseada. La secuencia de comandos EEM llama a este archivo a un desencadenador especificado y ejecuta todos los comandos enumerados en el archivo de texto.

Configuración y procedimiento paso a paso

Paso 1. Creación del archivo de texto con secuencia de comandos

1. Cree un archivo de texto que contenga los comandos necesarios en la secuencia deseada.
2. Copie el archivo en la memoria flash del dispositivo.
3. Utilice secuencias de comandos FTP, TFTP o TCL para crear o transferir el archivo.

Creación de un fichero de texto mediante secuencias de comandos TCL

Sintaxis:

```
<#root>
```

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

```
Enter Commands to Execute
```

```
}  
exit
```

Ejemplo de configuración:

Para crear una secuencia de comandos que apague la interfaz FiftyGigE1/1/0/3:

```
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#
```

Verifique el archivo en la memoria flash:

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

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

Contenido de COMMANDS_TO_EXECUTE.txt

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

Este archivo se puede crear mediante secuencias de comandos TCL o transferirse mediante FTP, TFTP o RCP. El archivo debe editarse según sea necesario para incluir cualquier secuencia de comandos requerida por el administrador de red.

Paso 2. Ejecute el archivo utilizando el script EEM

Creación de script EEM

Este applet EEM ejecuta todos los comandos del archivo especificado COMMANDS_TO_EXECUTE.

```
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
```

Explicación del Script EEM

++ applet del administrador de eventos Cisco_Live authorization bypass:

Nombre del applet del administrador de eventos: Registra un applet de EEM con el nombre especificado ('Cisco_Live' en este caso).

Omisión de autorización: Las palabras clave opcionales que especifican el applet deben omitir la autorización AAA.

`++ event none maxrun 1000:`

Evento ninguno: Esto especifica que la política de EEM se registra con el detector de eventos 'none', lo que significa que la política no es activada por ningún evento automático, sino que está pensada para ejecutarse manualmente.

Maxrun 1000: Este parámetro opcional establece el tiempo de ejecución máximo para el subprograma o la directiva en 1000 segundos. Si no se especifica, el tiempo de ejecución máximo predeterminado es de 20 segundos.

`++ action 020 cli command term exec prompt timestamp:`

Acción 020: Este es el identificador de etiqueta o secuencia para la acción dentro del script EEM. `term exec prompt timestamp` es un comando CLI que habilita las marcas de tiempo en el prompt en la sesión de terminal actual.

Esta característica ayuda en la solución de problemas y la auditoría proporcionando una referencia de tiempo para cada comando ejecutado.

La marca de tiempo aparece antes de la salida de los comandos, lo que facilita la correlación de eventos y ejecuciones de comandos con horas específicas.

`++ action 030 cli command term exec prompt expand:`

`term exec prompt expand` habilita el modo de prompt expandido para la sesión de terminal actual. Y configura el terminal para mostrar el mensaje completo sin truncamiento ni abreviatura.

Ayuda en secuencias de comandos o situaciones de solución de problemas en las que se necesita el contexto completo de aviso para mayor claridad o análisis.

El comando afecta solamente a la sesión de terminal actual y no persiste después de que la sesión finaliza.

`++ action 040 file open fileout1 flash:OUTPUT.txt a+:`

Archivo abierto: Especifica la operación para abrir un archivo.

Fileout1: Este es el descriptor de archivo o nombre de identificador utilizado en la secuencia de comandos para hacer referencia al archivo abierto.

Flash:OUTPUT.txt: Esta es la trayectoria y el nombre de archivo en el sistema de archivos flash donde reside o será creado el archivo.

a+: Este es el modo de acceso al archivo, lo que significa que el archivo se abre para leer y anexar. Si el archivo no existe, se creará. Los datos escritos en el archivo se agregarán al final sin truncar el contenido existente.

++ action 050 file open file in1 flash:COMMANDS_TO_EXECUTE.txt r:

Fileína1: Este es el descriptor de archivo o nombre de identificador utilizado en la secuencia de comandos para hacer referencia al archivo abierto.

Flash:COMMANDS_TO_EXECUTE.txt: Esta es la trayectoria y el nombre de archivo en el sistema de archivos flash donde reside el archivo.

r: Este es el modo de acceso al archivo, lo que significa que el archivo se abre sólo para lectura.

++ action 060 file read filein1 input:

Archivo leído: Especifica la operación que se va a leer de un archivo.

Fileína1: Este es el descriptor o identificador de archivo que se asignó cuando se abrió el archivo (por ejemplo, con la acción 050 file open file in1. Flash:COMMANDS_TO_EXECUTE.txt r).

Entrada: Este es el nombre de la variable donde se almacenarán los datos de lectura para su uso en acciones de script subsiguientes.

++ acción 070 para cada línea \$input '\n':

Foreach: Este comando especifica una iteración sobre una cadena.

Línea: Ésta es la variable iteradora que contendrá cada token (línea) durante cada iteración.

\$input: Ésta es la variable de cadena de entrada que contiene varias líneas o entradas.

'\n': Este es el delimitador utilizado para dividir la cadena de entrada en tokens, en este caso, el carácter de línea nueva.

Cada línea es un token y \n es el divisor en este caso.

++ action 080 cli command \$line:

Comando Cli: Especifica que la acción es ejecutar un comando CLI en el dispositivo.

\$line: Variable que contiene la cadena de comandos de CLI que se va a ejecutar. Las comillas aseguran que toda la cadena se trate como un solo comando.

++ acción 090 archivo write fileout1 '\$_cli_result':

Escritura de archivo: El comando EEM para escribir datos en un archivo.

Fileout1: El controlador de archivos previamente abierto por un comando como file open fileout1 flash:OUTPUT.txt a+, que representa el archivo en el que se escriben los datos.

'\$_cli_result': Variable que contiene el resultado del comando CLI o cualquier dato de cadena que se vaya a escribir en el archivo. Las comillas aseguran que todo el contenido se trate como una sola cadena.

Este comando se suele utilizar después de ejecutar comandos CLI y capturar su salida en \$_cli_result, lo que permite que la secuencia de comandos registre o archive la salida en un archivo en el almacenamiento flash del dispositivo.

++ acción 100 fin:

Finalizar: Palabra clave que significa el final de un bloque condicional como if, else o while.

++ action 120 file close file in1:

Cierre de archivo: Operación para cerrar el descriptor de archivo.

Fileína1: Identificador del descriptor de archivo que se va a cerrar, que se debe haber abierto anteriormente en la secuencia de comandos.

Ejecución de scripts EEM

Para accionar manualmente el script EEM:

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

En este ejemplo, la secuencia de comandos EEM se activa manualmente. El script puede tener cualquier disparador según los requisitos de la red.

Resultado de la ejecución del archivo

Ver el archivo de salida muestra los comandos ejecutados y sus resultados:

```
<#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

La interfaz FiftyGigE1/1/0/3 se cierra correctamente según la secuencia de comandos.

Summary

Este documento demostró cómo combinar el script TCL y el EEM de Cisco para crear un script EEM único y flexible capaz de ejecutar cualquier secuencia de comandos almacenada en un archivo de texto. Este enfoque permite a los administradores de red definir secuencias de comandos externamente y activar su ejecución automática o manualmente, lo que mejora la automatización y la eficacia operativa en los switches Cisco Catalyst que ejecutan Cisco IOS XE.

Acerca de esta traducción

Cisco ha traducido este documento combinando la traducción automática y los recursos humanos a fin de ofrecer a nuestros usuarios en todo el mundo contenido en su propio idioma.

Tenga en cuenta que incluso la mejor traducción automática podría no ser tan precisa como la proporcionada por un traductor profesional.

Cisco Systems, Inc. no asume ninguna responsabilidad por la precisión de estas traducciones y recomienda remitirse siempre al documento original escrito en inglés (insertar vínculo URL).