

Comprender las prácticas recomendadas y los guiones útiles para EEM

Contenido

[Introducción](#)

[Prerequisites](#)

[Requirements](#)

[Componentes Utilizados](#)

[Convenciones](#)

[Mejores medidas](#)

[Confirmar que la autenticación adecuada está en vigor](#)

[Agregar restricciones para EEM Runtime y Rate-limit](#)

[Evitar la ejecución fuera de servicio](#)

[Deshabilitar paginación](#)

[Guiones de diseño para el mantenimiento futuro](#)

[Patrones lógicos comunes de EEM](#)

[Rutas de código de sucursal con if/else](#)

[Sentencias de Loop Over](#)

[Extraer salida mediante expresiones regulares \(Regex\)](#)

[Scripts EEM útiles](#)

[Seguimiento de Dirección MAC Específica para Aprendizaje de Dirección MAC](#)

[Supervisión de CPU alta mediante OID SNMP](#)

[Coincidencia dinámica de un PID y salida de la pila de registros](#)

[Actualización de un switch](#)

[Volcar datos de diagnóstico en un archivo cuando un objeto de seguimiento de SLA IP deja de funcionar](#)

[Enviar un correo electrónico desde EEM](#)

[Cierre de un puerto según lo programado](#)

[Apagar una interfaz si se alcanza una velocidad de paquetes por segundo \(PPS\) determinada](#)

[Referencias](#)

Introducción

Este documento describe las prácticas recomendadas de configuración de scripts de Embedded Event Manager (EEM) en dispositivos Cisco IOS®-XE y proporciona ejemplos de sintaxis común y scripts útiles.

Prerequisites

Requirements

Este documento asume que el lector ya está familiarizado con la función Embedded Event

Manager (EEM) de Cisco IOS®/IOS-XE. Si todavía no está familiarizado con esta función, por favor lea la [Descripción General de la Función EEM](#) primero.

Componentes Utilizados

La información que contiene este documento se basa en las siguientes versiones de software y hardware.

- Switches Cisco Catalyst 9300, 9400 y 9500 que ejecutan la versión 16.X o 17.X del software Cisco IOS

Estos scripts no son compatibles con Cisco TAC y se proporcionan tal cual con fines educativos.

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. Si tiene una red en vivo, asegúrese de entender el posible impacto de cualquier comando.

Convenciones

Consulte [Convenciones de Consejos Técnicos de Cisco para obtener información sobre las convenciones sobre documentos](#).

Mejores medidas

Esta sección cubre algunos de los problemas más comunes observados con el diseño e implementación de scripts EEM. Para obtener información adicional sobre las prácticas recomendadas de EEM, consulte el documento Prácticas recomendadas de EEM al que se hace referencia en la sección Referencias.

Confirmar que la autenticación adecuada está en vigor

Si su dispositivo utiliza AAA, debe asegurarse de que los scripts EEM configurados en el dispositivo estén configurados con un usuario AAA capaz de ejecutar los comandos en el script, o que la omisión de autorización esté configurada con el comando **authorization bypass** en la definición del script.

Agregar restricciones para EEM Runtime y Rate-limit

De forma predeterminada, las secuencias de comandos de EEM pueden ejecutarse durante un máximo de 20 segundos. Si diseña una secuencia de comandos que tarda más en ejecutarse o que tiene que esperar entre la ejecución del comando, especifique un valor **maxrun** en el desencadenador de eventos del subprograma para modificar el temporizador de ejecución predeterminado.

También es importante tener en cuenta la frecuencia con la que se puede ejecutar el evento que desencadena el script EEM. Si activa un script a partir de una condición que se produce rápidamente en un corto período de tiempo (por ejemplo, el desencadenador syslog para las inestabilidad MAC), es importante incluir una condición de límite de velocidad en el script EEM para evitar un número excesivo de ejecuciones en paralelo y evitar el agotamiento de los recursos del dispositivo.

Evitar la ejecución fuera de servicio

Como se describe en la documentación de EEM, el orden de ejecución de las sentencias `action` se controla mediante su etiqueta (por ejemplo, `action 0001 cli command enable` tiene una etiqueta 0001). Este valor de etiqueta NO es un número, sino un valor alfanumérico. Las acciones se ordenan en secuencia de clave alfanumérica ascendente, utilice el argumento `label` como clave de ordenación y se ejecutan en esta secuencia. Esto puede dar lugar a un orden de ejecución inesperado, en función de la estructura de las etiquetas de acción.

Tenga en cuenta este ejemplo:

```
event manager applet test authorization bypass
event timer watchdog time 60 maxrun 60
action 13 syslog msg "You would expect to see this message first"
action 120 syslog msg "This message prints first"
```

Dado que 120 es anterior a 13 en una comparación alfanumérica, este script no se ejecuta en el orden esperado. Para evitar esto, es útil utilizar un sistema de relleno como este:

```
event manager applet test authorization bypass
event timer watchdog time 60 maxrun 60
action 0010 syslog msg "This message appears first"
action 0020 syslog msg "This message appears second"
action 0120 syslog msg "This message appears third"
```

Debido al relleno aquí, las instrucciones numeradas se evalúan en el orden esperado. El incremento de 10 entre cada etiqueta permite que las sentencias adicionales se inserten en el script EEM más adelante donde sea necesario, sin necesidad de volver a numerar todas las sentencias subsiguientes.

Deshabilitar paginación

EEM busca la indicación del dispositivo para determinar cuándo se completa la salida del comando. Los comandos que emiten más datos de los que se pueden mostrar en una pantalla (según la configuración de la longitud del terminal), pueden impedir que se completen los scripts EEM (y que finalmente se eliminen a través del temporizador `maxrun`) ya que el mensaje del dispositivo no se muestra hasta que se vean todas las páginas de la salida. Configure el **término** `len 0` al inicio de los scripts EEM que examinan salidas grandes.

Guiones de diseño para el mantenimiento futuro

Cuando diseñe un script EEM, deje espacios entre las etiquetas de acción para facilitar la actualización de la lógica del script EEM en el futuro. Cuando se dispone de espacios adecuados (es decir, dos sentencias como `action 0010` y `action 0020` dejan un espacio de 9 etiquetas que se pueden insertar), se pueden agregar nuevas sentencias según sea necesario sin volver a numerar o verificar las etiquetas de acción y asegurarse de que las acciones continúan ejecutándose en el orden esperado.

Hay comandos comunes que debe ejecutar al principio de los scripts EEM. Esto puede incluir:

- `set terminal length to 0`
- entrar en modo de activación
- `enable automatic timestamp for command output`

Este es un patrón común en los ejemplos mostrados en este documento, donde muchas de las secuencias de comandos comienzan con las mismas 3 sentencias action para configurarlo.

Patrones lógicos comunes de EEM

Esta sección trata algunos patrones lógicos y bloques de sintaxis comunes utilizados en scripts EEM. Los ejemplos aquí no son scripts completos, sino demostraciones de cómo se puede utilizar una funcionalidad específica para crear scripts EEM complejos.

Rutas de código de sucursal con if/else

Las variables EEM se pueden utilizar para controlar el flujo de ejecución de scripts EEM. Tenga en cuenta este guion de EEM:

```
event manager applet snmp_cpu authorization bypass
event timer watchdog time 60
action 0010 info type snmp oid 10.10.10.1.4.1.9.9.109.1.1.1.1.3.1 get-type exact
action 0020 if $_info_snmp_value ge "50"
action 0030 syslog msg "This syslog message is sent if CPU utilization is above 50%"
action 0040 elseif $_info_snmp_value ge "30"
action 0050 syslog msg "This syslog message is sent if CPU utilization is above 30% and below
50%"
action 0060 else
action 0070 syslog msg "This syslog message is sent if CPU utilization is below 30%"
action 0080 end
```

Este script se ejecuta cada minuto. Examine el valor del OID de SNMP para la utilización de la CPU y, a continuación, introduzca una de las tres rutas de ejecución diferentes basadas en el valor del OID. Se pueden utilizar sentencias similares en cualquier otra variable EEM legal para generar flujos de ejecución complejos en scripts EEM.

Sentencias de Loop Over

Los loops de ejecución se pueden utilizar para acortar significativamente los scripts EEM y para que sea más fácil razonar sobre ellos. Considere esta secuencia de comandos, diseñada para extraer las estadísticas de la interfaz para Te2/1/15 6 veces durante un período de 1 minuto para verificar si hay períodos pequeños de alta utilización:

```
event manager applet int_util_check auth bypass
event timer watchdog time 300 maxrun 120
action 0001 cli command "enable"
action 0002 cli command "term exec prompt timestamp"
action 0003 cli command "term length 0"
action 0010 syslog msg "Running iteration 1 of command"
action 0020 cli command "show interface te2/1/15 | append flash:interface_util.txt"
action 0030 wait 10
action 0040 syslog msg "Running iteration 2 of command"
action 0050 cli command "show interface te2/1/15 | append flash:interface_util.txt"
action 0060 wait 10
action 0070 syslog msg "Running iteration 3 of command"
action 0080 cli command "show interface te2/1/15 | append flash:interface_util.txt"
action 0090 wait 10
action 0100 syslog msg "Running iteration 4 of command"
action 0110 cli command "show interface te2/1/15 | append flash:interface_util.txt"
action 0120 wait 10
```

```
action 0130 syslog msg "Running iteration 5 of command"
action 0140 cli command "show interface te2/1/15 | append flash:interface_util.txt"
action 0150 wait 10
action 0160 syslog msg "Running iteration 6 of command"
action 0170 cli command "show interface te2/1/15 | append flash:interface_util.txt"
```

Con las construcciones de loop EEM, este script se puede acortar significativamente:

```
event manager applet int_util_check auth bypass
event timer watchdog time 300 maxrun 120
action 0001 cli command "enable"
action 0002 cli command "term exec prompt timestamp"
action 0003 cli command "term length 0"
action 0010 set loop_iteration 1
action 0020 while $loop_iteration le 6
action 0030 syslog msg "Running iteration $loop_iteration of command"
action 0040 cli command "show interface te2/1/15 | append flash:interface_util.txt"
action 0050 wait 10
action 0060 increment loop_iteration 1
action 0070 end
```

Extraer salida mediante expresiones regulares (Regex)

La instrucción regexp de EEM se puede utilizar para extraer valores de la salida del comando para utilizarlos en comandos posteriores y permitir la creación de comandos dinámicos dentro de la propia secuencia de comandos de EEM. Consulte este bloque de código para obtener un ejemplo para extraer el PID del MOTOR SNMP de la salida de `show proc cpu | i motor SNMP` e imprimirlo en un mensaje syslog. Este valor extraído también se puede utilizar en otros comandos que requieren un PID para ejecutarse.

```
event manager applet check_pid auth bypass
event none
action 0010 cli command "show proc cpu | i SNMP ENGINE"
action 0020 regexp "^[ ]*([0-9]+) .*" $_cli_result match match1
action 0030 syslog msg "Found SNMP Engine PID $match1"
```

Scripts EEM útiles

Seguimiento de Dirección MAC Específica para Aprendizaje de Dirección MAC

En este ejemplo, se realiza un seguimiento de la dirección MAC **b4e9.b0d3.6a41**. El script verifica cada 30 segundos para ver si la dirección MAC especificada se ha aprendido en las tablas ARP o MAC. Si se ve el MAC, el script realiza estas acciones:

- muestra un mensaje de syslog (esto es útil cuando desea confirmar dónde se aprende una dirección MAC, o cuándo/con qué frecuencia se aprende).

Instrumentación

```
event manager applet mac_trace authorization bypass
event timer watchdog time 30
action 0001 cli command "enable"
action 0002 cli command "term exec prompt timestamp"
action 0003 cli command "term length 0" action 0010 cli command "show ip arp | in
b4e9.b0d3.6a41" action 0020 regexp ".*(ARPA).*" $_cli_result action 0030 if $_regexp_result eq 1
action 0040 syslog msg $_cli_result action 0050 end action 0060 cli command "show mac add vlan 1
```

```
| in b4e9.b0d3.6a41" action 0070 regexp ".*(DYNAMIC).*" $_cli_result action 0080 if
$_regexp_result eq 1 action 0090 syslog msg $_cli_result action 0100 end
```

Supervisión de CPU alta mediante OID SNMP

Esta secuencia de comandos controla un OID SNMP utilizado para leer el porcentaje de ocupación de la CPU en los últimos 5 segundos. Cuando la CPU está ocupada en más del 80%, el script realiza estas acciones:

- crea una marca de tiempo a partir de la salida de show clock y la utiliza para crear un nombre de archivo único
- los resultados sobre el proceso y el estado del software se escriben en este archivo
- se configura una captura de paquetes integrada (EPC) para capturar 10 segundos de tráfico destinados al plano de control y escribirlos en un archivo.
- una vez que se completa la captura EPC, se elimina la configuración EPC y se cierra el script.

Instrumentación

```
event manager applet high-cpu authorization bypass
event snmp oid 10.10.10.1.4.1.9.9.109.1.1.1.1.3.1 get-type next entry-op gt entry-val 80 poll-
interval 1 ratelimit 300 maxrun 180
action 0001 cli command "enable"
action 0002 cli command "term exec prompt timestamp"
action 0003 cli command "term length 0"
action 0010 syslog msg "High CPU detected, gathering system information."
action 0020 cli command "show clock"
action 0030 regex "([0-9]|[0-9][0-9]):([0-9]|[0-9][0-9]):([0-9]|[0-9][0-9])" $_cli_result match
match1
action 0040 string replace "$match" 2 2 "."
action 0050 string replace "$_string_result" 5 5 "."
action 0060 set time $_string_result
action 0070 cli command "show proc cpu sort | append flash:tac-cpu-$time.txt"
action 0080 cli command "show proc cpu hist | append flash:tac-cpu-$time.txt"
action 0090 cli command "show proc cpu platform sorted | append flash:tac-cpu-$time.txt"
action 0100 cli command "show interface | append flash:tac-cpu-$time.txt"
action 0110 cli command "show interface stats | append flash:tac-cpu-$time.txt"
action 0120 cli command "show log | append flash:tac-cpu-$time.txt"
action 0130 cli command "show ip traffic | append flash:tac-cpu-$time.txt"
action 0140 cli command "show users | append flash:tac-cpu-$time.txt"
action 0150 cli command "show platform software fed switch active punt cause summary | append
flash:tac-cpu-$time.txt"
action 0160 cli command "show platform software fed switch active cpu-interface | append
flash:tac-cpu-$time.txt"
action 0170 cli command "show platform software fed switch active punt cpuq all | append
flash:tac-cpu-$time.txt"
action 0180 cli command "no monitor capture tac_cpu"
action 0190 cli command "monitor capture tac_cpu control-plane in match any file location
flash:tac-cpu-$time.pcap"
action 0200 cli command "monitor capture tac_cpu start" pattern "yes"
action 0210 cli command "yes"
action 0220 wait 10
action 0230 cli command "monitor capture tac_cpu stop"
action 0240 cli command "no monitor capture tac_cpu"
```

Coincidencia dinámica de un PID y salida de la pila de registros

Esta secuencia de comandos busca un mensaje de syslog que indique que la cola de entrada SNMP está llena y realiza estas acciones:

- registra el resultado de **show proc cpu sort** en un archivo
- extrae el PID del proceso SNMP ENGINE mediante regex
- utiliza el PID SNMP en los comandos subsiguientes para obtener los datos de pila para el PID
- quita el script de la configuración para que no se produzcan más ejecuciones del mismo

Instrumentación

```
event manager applet TAC-SNMP-INPUT-QUEUE-FULL authorization bypass
event syslog pattern "INPUT_QFULL_ERR" ratelimit 40 maxrun 120
action 0010 cli command "en"
action 0020 cli command "show proc cpu sort | append flash:TAC-SNMP.txt"
action 0030 cli command "show proc cpu | i SNMP ENGINE"
action 0040 regexp "^[ ]*([0-9]+) .*" $_cli_result match match1
action 0050 syslog msg "Found SNMP Engine PID $match1"
action 0060 cli command "show stacks $match1 | append flash:TAC-SNMP.txt"
action 0070 syslog msg "$_cli_result"
action 0080 cli command "configure terminal"
action 0090 cli command "no event manager applet TAC-SNMP-INPUT-QUEUE-FULL"
action 0100 cli command "end"
```

Actualización de un switch

Esta secuencia de comandos se configura para que coincida con el patrón en el mensaje no estándar devuelto por el comando **install add file <file> activate commit** y responda a los mensajes. No se configura ningún evento desencadenador, por lo que el usuario debe activar manualmente el script EEM cuando la actualización deba realizarse mediante **event manager run UPGRADE**. El temporizador maxrun se configura para 300 segundos en lugar del valor predeterminado de 20 segundos, ya que el comando **install add** tarda una cantidad significativa de tiempo en ejecutarse.

Instrumentación

```
event manager applet UPGRADE authorization bypass
event none maxrun 300
action 0001 cli command "enable"
action 0002 cli command "term length 0" action 0020 cli command "install add file
flash:cat9k_iosxe.16.06.02.SPA.bin activate commit" pattern "y\n" action 0030 cli command "y"
pattern "y\n" action 0040 syslog msg "Reloading device to upgrade code" action 0050 cli command
"y"
```

Volcar datos de diagnóstico en un archivo cuando un objeto de seguimiento de SLA IP deja de funcionar

Este script se activa cuando el objeto IP SLA 11 deja de funcionar y realiza estas acciones:

- Recopilar tabla MAC, tabla ARP, registros del sistema y tabla de enrutamiento
- Escribir información en un archivo en flash: llamado sla_track.txt

Instrumentación

```
ip sla 10
icmp-echo 10.10.10.10 source-ip 10.10.10.10
frequency 10
exit
ip sla schedule 10 life forever start-time now
track 11 ip sla 10 reachability
```

```
exit
event manager applet track-10 authorization bypass
event track 11 state down
action 0001 cli command "enable"
action 0002 cli command "term exec prompt timestamp"
action 0003 cli command "term length 0" action 0010 syslog msg "IP SLA object 10 has gone down"
action 0020 cli command "show mac address-table detail | append flash:sla_track.txt" action 0030
cli command "show ip arp | append flash:sla_track.txt" action 0040 cli command "show log |
append flash:sla_track.txt" action 0050 cli command "show ip route | append flash:sla_track.txt"
```

Enviar un correo electrónico desde EEM

Esta secuencia de comandos se activa cuando se ve el patrón descrito en la sentencia de patrón syslog del evento, y realiza estas acciones:

- envía un correo electrónico desde un servidor de correo electrónico interno (se supone que el servidor de correo electrónico interno permite la autenticación abierta desde el dispositivo).

Instrumentación

```
event manager environment email_from email_address@company.test
event manager environment email_server 192.168.1.1
event manager environment email_to dest_address@company.test
event manager applet email_syslog
event syslog pattern "SYSLOG PATTERN HERE" maxrun 60
action 0010 info type routername
action 0020 mail server "$email_server" to "$email_to" from "$email_from" subject "SUBJECT OF
EMAIL - Syslog seen on $_info_routername" body "BODY OF YOUR EMAIL GOES HERE"
```

Cierre de un puerto según lo programado

Este script apaga el puerto Te2/1/15 todos los días a las 6PM.

Instrumentación

```
event manager applet shut_port authorization bypass
event timer cron cron-entry "0 18 * * *"
action 0001 cli command "enable"
action 0002 cli command "term exec prompt timestamp"
action 0003 cli command "term length 0" action 0010 syslog msg "shutting port Te2/1/15 down"
action 0030 cli command "config t" action 0040 cli command "int Te2/1/15" action 0050 cli
command "shutdown" action 0060 cli command "end"
```

Apagar una interfaz si se alcanza una velocidad de paquetes por segundo (PPS) determinada

Este script verifica la velocidad PPS en la interfaz Te2/1/9 en la dirección TX cada segundo. Si la velocidad de PPS supera los 100, realiza estas acciones:

- registra la salida **show int** para la interfaz en syslog
- apaga la interfaz

Instrumentación

```
event manager applet disable_link authorization bypass
event interface name te2/1/9 parameter transmit_rate_pps entry-op ge entry-val 100 poll-
interval 1 entry-type value
```

```
action 0001 cli command "enable"  
action 0002 cli command "term length 0" action 0010 syslog msg "Detecting high input rate on  
interface te2/1/9. Shutting interface down." action 0020 cli command "show int te2/1/9" action  
0030 syslog msg $_cli_result action 0040 cli command "config t" action 0050 cli command "int  
te2/1/9" action 0060 cli command "shutdown" action 0070 cli command "end"
```

Referencias

- [Prácticas recomendadas de Cisco EEM](#)

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).