Compreender as práticas recomendadas e os scripts úteis para o EEM

Contents

Introdução

Este documento descreve as práticas recomendadas de configuração de script do Embedded Event Manager (EEM) em dispositivos Cisco IOS® XE.

Pré-requisitos

Requisitos

A Cisco recomenda que você tenha conhecimento e familiaridade com este tópico:

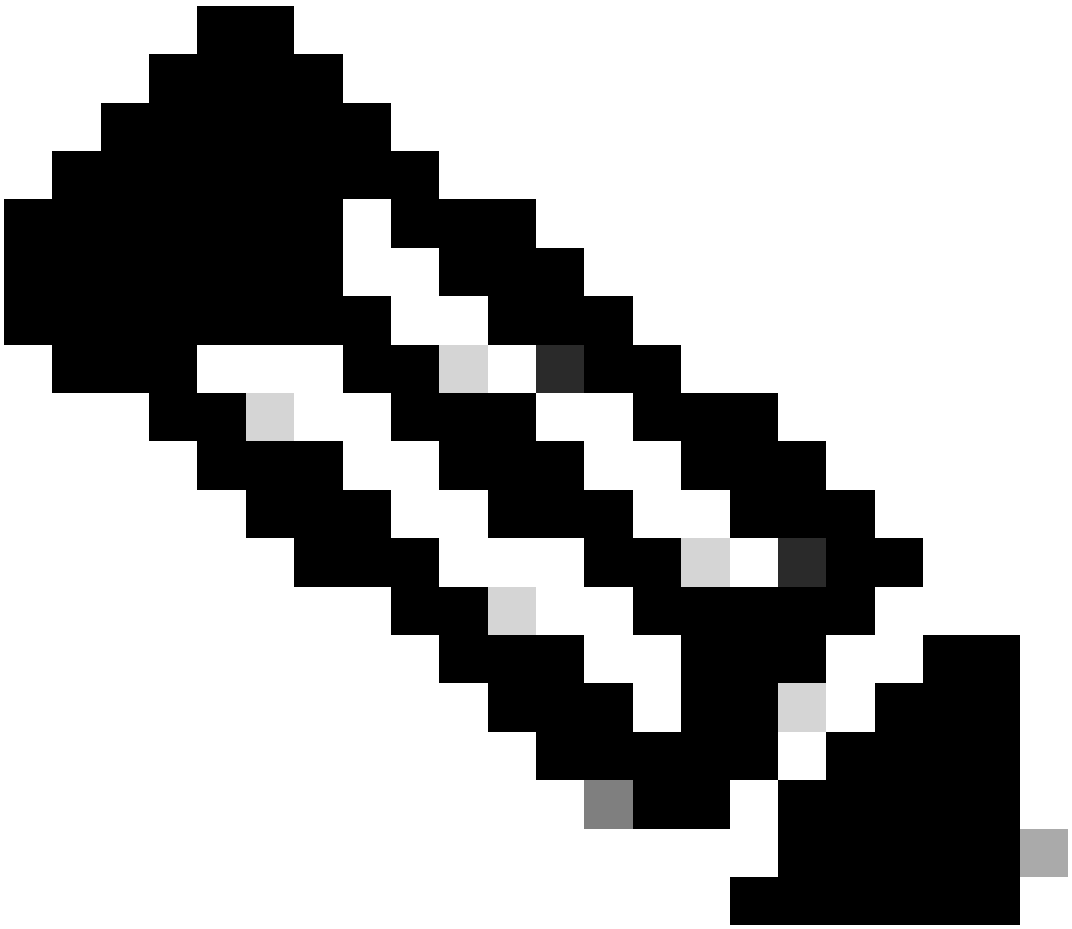
- Cisco IOS e Cisco IOS XE Embedded Event Manager (EEM)

Se você ainda não estiver familiarizado com esse recurso, leia a Visão geral do recurso EEM primeiro.

Componentes Utilizados

As informações neste documento são baseadas nestas versões de software e hardware:

- Switches Cisco Catalyst 9300, 9400 e 9500
- Software Cisco IOS versão 16.X ou 17.X



> Observação: esses scripts não são suportados pelo Cisco TAC e são fornecidos no estado em que se encontram para fins educacionais.

As informações neste documento foram criadas a partir de dispositivos em um ambiente de laboratório específico. Todos os dispositivos utilizados neste documento foram iniciados com uma configuração (padrão) inicial. Se a rede estiver ativa, certifique-se de que você entenda o impacto

potencial de qualquer comando.

Conventions

Consulte as Convenções de Dicas Técnicas da Cisco para obter informações sobre convenções de documentos.

Melhores práticas

Esta seção aborda alguns dos problemas mais comuns observados com o projeto e a implementação de scripts EEM. Para obter informações adicionais sobre as práticas recomendadas do EEM, consulte o documento de práticas recomendadas do EEM mencionado na seção Referências.

Confirme se a autenticação apropriada está em vigor

Se o dispositivo usar AAA, você deverá garantir que os scripts EEM configurados no dispositivo estejam configurados com um usuário AAA capaz de executar os comandos no script ou que o desvio de autorização esteja configurado com o comando authorization bypass na definição do script.

Adicionar Restrições para Tempo de Execução EEM e Limite de Taxa

Por padrão, os scripts EEM podem ser executados por no máximo 20 segundos. Se você criar um script que leve mais tempo para ser executado ou que precise aguardar entre a execução do comando, especifique um valor maxrun no disparador de eventos do applet para alterar o temporizador de execução padrão.

Também é importante considerar com que frequência o evento que aciona o script EEM pode ser executado. Se você acionar um script a partir de uma condição que ocorra rapidamente em um curto período de tempo (por exemplo, disparador de syslog para oscilações de MAC), é importante incluir uma condição de limite de taxa no script EEM para evitar um número excessivo de execuções paralelas e impedir o esgotamento dos recursos do dispositivo.

Evite a execução fora de serviço

Conforme descrito na documentação do EEM, a ordem de execução das instruções de ação é controlada por seu rótulo (por exemplo, o comando de cli enable da ação 0001 tem um rótulo de 0001). Este valor de rótulo NÃO é um número, mas sim alfanumérico. As ações são classificadas em sequência de chave alfanumérica crescente, usam o argumento label como chave de classificação e são executadas nessa sequência. Isso pode levar a uma ordem inesperada de execução, com base em como você estrutura seus rótulos de ação.

Considere este exemplo:

```
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"
```

Como 120 é anterior a 13 em uma comparação alfanumérica, este script não é executado na ordem esperada. Para evitar isso, é útil usar um sistema de preenchimento 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"
```

Devido ao preenchimento aqui, as instruções numeradas são avaliadas na ordem esperada. O incremento de 10 entre cada rótulo permite que instruções adicionais sejam inseridas no script EEM posteriormente, onde necessário, sem a necessidade de renumerar todas as instruções subsequentes.

Desabilitar paginação

O EEM procura o prompt do dispositivo para determinar quando a saída do comando está completa. Os comandos que geram mais dados do que podem ser exibidos em uma tela (conforme configurado pelo comprimento do terminal) podem impedir que os scripts EEM sejam concluídos (e eventualmente eliminados através do temporizador maxrun), pois o prompt do dispositivo não é mostrado até que todas as páginas da saída sejam visualizadas. Configure o termo len 0 no início dos scripts EEM que examinam saídas grandes.

Scripts de design para futura capacidade de manutenção

Ao projetar um script EEM, deixe lacunas entre os rótulos de ação para facilitar a atualização da lógica do script EEM no futuro. Quando lacunas apropriadas estão disponíveis (isto é, duas instruções como ação 0010 e ação 0020 deixam uma lacuna de 9 rótulos que podem ser inseridos), novas instruções podem ser adicionadas conforme necessário sem renumerar ou reverificar os rótulos de ação e garantir que as ações continuem a ser executadas na ordem esperada.

Há comandos comuns que você precisa executar no início de seus scripts EEM. Isso pode incluir:

- definir o comprimento do terminal como 0
- insira o modo enable
- habilitar carimbo de data/hora automático para saída de comando

Esse é um padrão comum nos exemplos mostrados neste documento, em que muitos dos scripts começam com as mesmas 3 instruções de ação para configurar isso.

Padrões lógicos comuns do EEM

Esta seção aborda alguns padrões lógicos comuns e blocos de sintaxe usados em scripts EEM. Os exemplos aqui não são scripts completos, mas demonstrações de como a funcionalidade específica pode ser usada para criar scripts EEM complexos.

Caminhos de código de ramificação com If/Else

As variáveis EEM podem ser usadas para controlar o fluxo de execução de scripts EEM. Considere este script EEM:

```
event manager applet snmp_cpu authorization bypass
event timer watchdog time 60
action 0010 info type snmp oid 1.3.6.1.4.1.9.9.109.1.1.1.1.3 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
```

Esse script é executado a cada minuto. Examine o valor do OID de SNMP para utilização da CPU e insira um dos três caminhos de execução diferentes com base no valor do OID. Instruções semelhantes podem ser usadas em qualquer outra variável EEM legal para criar fluxos de execução complexos em scripts EEM.

Instruções de Loop

Os loops de execução podem ser usados para encurtar significativamente os scripts EEM e torná-los mais fáceis de serem discutidos. Considere este script, projetado para receber 6 vezes as estatísticas de interface para Te2/1/15 durante um período de 1 minuto para verificar pequenos períodos de alta utilização:

```
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"
```

Com construções de loop EEM, este script pode ser significativamente reduzido:

```
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
```

Extrair saída através de expressões regulares (Regex)

A instrução EEM regexp pode ser usada para extrair valores da saída do comando a serem usados em comandos subsequentes e permitir a criação dinâmica de comandos dentro do próprio script EEM. Consulte este bloco de código para obter um exemplo para extrair o SNMP ENGINE PID da saída de show proc cpu | i SNMP engine e imprimi-lo em uma mensagem de syslog. Esse valor extraído também pode ser usado em outros comandos que exigem um PID para serem executados.

```
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 úteis

Acompanhar endereço MAC específico para aprendizagem de endereço MAC

Neste exemplo, o endereço MAC b4e9.b0d3.6a41 é rastreado. O script verifica a cada 30 segundos se o endereço MAC especificado foi aprendido nas tabelas ARP ou MAC. Se o MAC for

visto, o script executará estas ações:

- gera uma mensagem de syslog (isso é útil quando você deseja confirmar onde um endereço MAC é aprendido ou quando/com que frequência ele é aprendido).

Implementação

```
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
```

Monitorar Alta CPU via SNMP OID

Este script monitora um OID de SNMP usado para ler a porcentagem de ocupação da CPU nos últimos 5 segundos. Quando a CPU está acima de 80% ocupada, o script executa estas ações:

- cria um carimbo de data/hora a partir da saída de show clock e o utiliza para criar um nome de arquivo exclusivo
- as saídas sobre o processo e o estado do software são gravadas neste arquivo
- Um EPC (Embedded Packet Capture) é configurado para capturar 10 segundos de tráfego destinado ao plano de controle e gravá-lo em um arquivo.
- depois que a captura EPC é concluída, a configuração do EPC é removida e o script é encerrado.

Implementação

```
event manager applet high-cpu authorization bypass
event snmp oid 1.3.6.1.4.1.9.9.109.1.1.1.1.3 get-type next entry-op gt entry-val 80 poll-interval 1 rat
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
action 0160 cli command "show platform software fed switch active cpu-interface | append flash:tac-cpu-
action 0170 cli command "show platform software fed switch active punt cpuq all | append flash:tac-cpu-
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
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"
```

Corresponder dinamicamente um PID e registrar a saída da pilha

Este script procura uma mensagem de syslog informando que a fila de entrada SNMP está cheia e executa estas ações:

- registra a saída de show proc cpu sort em um arquivo
- extrai o PID do processo SNMP ENGINE via regex
- usa o SNMP PID em comandos subsequentes para obter os dados da pilha para o PID
- remove o script da configuração para que não ocorram mais execuções dele

Implementação

```
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"
```

Atualizar um switch

Esse script é configurado para correspondência de padrão no prompt fora do padrão retornado pelo comando install add file <file> ativate commit e responde aos prompts. Nenhum evento disparador está configurado, portanto, o script EEM deve ser acionado manualmente por um usuário quando a atualização precisa ocorrer por meio da execução de ATUALIZAÇÃO do gerenciador de eventos. O temporizador maxrun é definido para 300 segundos em vez do valor

padrão de 20 segundos, pois o comando install add leva um tempo significativo para ser executado.

Implementação

```
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 "
action 0030 cli command "y" pattern "y\/n"
action 0040 syslog msg  "Reloading device to upgrade code"
action 0050 cli command "y"
```

Despejar dados de diagnóstico em um arquivo quando um objeto IP rastreado de SLA cair

Esse script é acionado quando o objeto IP SLA 11 é desativado e executa estas ações:

- Coletar tabela MAC, tabela ARP, syslogs e tabela de Roteamento
- Gravar informações em um arquivo em flash: chamado sla_track.txt

Implementação

```
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 um e-mail do EEM

Esse script é acionado quando o padrão descrito na instrução de padrão syslog de evento é visto e executa estas ações:

- Envia um email de um servidor de email interno (isso pressupõe que o servidor de email

interno permita a autenticação aberta do dispositivo).

Implementação

```
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 - S
```

Desligar uma porta de acordo com um agendamento

Este script desliga a porta Te2/1/15 todos os dias às 18h.

Implementação

```
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"
```

Desligar uma interface se uma taxa de pacotes por segundo (PPS) for atingida

Este script verifica a taxa PPS na interface Te2/1/9 na direção TX a cada segundo. Se a taxa de PPS exceder 100, ele executará as seguintes ações:

- Registra a saídashow int da interface no syslog.
    - Desliga a interface.

Implementação

```
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"
```

Informações Relacionadas

- **Práticas recomendadas do Cisco EEM**
- **Suporte técnico e downloads da Cisco**

Sobre esta tradução
A Cisco traduziu este documento com a ajuda de tecnologias de tradução automática e humana para oferecer conteúdo de suporte aos seus usuários no seu próprio idioma, independentemente da localização.

Observe que mesmo a melhor tradução automática não será tão precisa quanto as realizadas por um tradutor profissional.

A Cisco Systems, Inc. não se responsabiliza pela precisão destas traduções e recomenda que o documento original em inglês (link fornecido) seja sempre consultado.