EEM Scripts used to Troubleshoot

Document ID: 116176

Contributed by Cisco TAC Engineers.
Oct 10, 2013

Contents

Introduction

This document describes how to use the Embedded Event Manager (EEM) tool to troubleshoot issues on the network that are otherwise hard to pin point or do not have a regular frequency which allows normal troubleshooting.

What is EEM?

EEM is a flexible system designed to customize Cisco IOS®, XR, and NX−OS. EEM allows you to automate tasks, perform minor enhancements, and create workarounds.

When to use EEM?

EEM scripts have two purposes:

- To help troubleshoot an issue − When you need to troubleshoot problems of an intermittent nature, EEM scripts can be particularly useful. They allow you to automate the collection process of ***show*** command outputs and ***debug*** commands which allows you to capture data that would otherwise be extremely hard to gather.
- To help provide a solution − In cases where a temporary workaround is required while the Technical Assistance Center (TAC) does a root cause analysis. Take for example a situation where the problem is intermittent, but the reset of an interface fixes the problem. EEM scripts can be used to trigger this action as soon as the problem begins.

In either case, in order to use EEM scripts it is necessary to identify a trigger event which you can then use to trigger the script.

Sample EEM Scripts

Sample EEM scripts to help troubleshoot issues are included in this section.

Run Commands Every 30 Seconds to File in Flash

Run commands every 30 seconds to the file in flash; ***show*** commands can be adapted to whatever you want to run:

```
event manager applet show-rtp-streams
  event timer watchdog name timer time 30
  action 0.5 cli command "enable"
  action 1.0 cli command "show clock | append flash:filename.txt"
  action 2.0 cli command "show ip cache flow | append flash:filename.txt"
  action 3.0 cli command "show voip rtp conn" | append flash:filename.txt"
  action 4.0 cli command "show call active voice br" | append flash:filename.txt"
```

Note: Some flash file systems (such as bootflash or slotX) do not support the append operation. This is not an EEM limitation, but rather a limitation of linear flash file systems. Random access file systems such as diskX or the flash on desktop switches (3560, 3750, and so on) support appending to a file.

Note: If the device is configured for Authentication, Authorization, and Accounting (AAA) command authorization, EEM needs to be configured with a username that is authorized to run all of the CLI commands in all of the configured EEM policies. In order to do this, enter ***event manager session cli username USER*** where USER is the appropriate AAA user.

Use a Watchdog to Dump the Router Log Every X Seconds to an FTP Server or Flash

Similarly, you can use a watchdog in order to dump the router log every X seconds to an FTP server or flash:

```
event manager applet dump-log
  event timer watchdog name timer time 1800
  action 0.5 cli command "enable"
  action 1.0 cli command "show log | append ftp://user:pass@10.1.1.1/debugs.txt"
```

You can also use an Expect script on a UNIX device inside a cronjob to pull the log every X minutes. Instead of pushing it with EEM; replace the username and password with the proper strings for login credentials:

```
 > dhcp-64-102-154-159:Desktop sholl$ cat login-script
 > #!/usr/bin/expect
 >
 > set timeout 60
 > spawn telnet -N 10.1.1.1
 >
 > # Uncomment these if you are prompted for a username by the router
 > # expect "login:"
 > # send "username\n"
 > expect "Password: "
 > send "password\n"
 > expect ">"
 > send "en\n"
 > expect "Password:"
 > send "password\n"
 > expect "#"
 > send "term len 0\n"
 > expect "#"
 > send "sh log\n"
```

```
> expect "#"
> send "exit\n"
> send "exit\n"
>
> dhcp-64-102-154-159:Desktop sholl$ crontab -e
>
> # min   hour    mday    month   wday    command
>    0     4       0        0       0     Desktop/login-script >> outputlog.txt
```

Enable Debugs at Boot of Router

Enable debugs at the boot of the router; change the debugs to whatever you want to enable:

```
event manager applet en-debugs-at-boot
event timer cron cron-entry "@reboot"
  action 1.0 cli command "enable"
  action 2.0 cli command "debug isdn q931"
  action 2.2 cli command "debug isdn q921"
  action 2.4 cli command "debug isdn standard"
```

Disable Debugs after a Specific Message

Disable debugs after a specific debug message is detected in order to prevent the log from filling up. Disable the EEM script (itself) afterwards. Change the pattern to what matches your situation:

```
event manager applet disableDebugsOnError
  event syslog occurs 1 pattern "Endpt not available"
  action 3.0 cli command "enable"
  action 3.2 cli command "un all"
  action 3.3 cli command "config t"
  action 3.4 cli command "no event manager applet disableDebugsOnError"
  action 3.5 cli command "end"
```

Troubleshoot

Troubleshoot VPN Issues

While most forms of VPN issues do not generally require EEM in order to troubleshoot, in some cases the problem can be transient which makes it hard to obtain the necessary information. Relevant cases inlcude:

- Utilize EEM to Troubleshoot IGP Flaps/Outage over VPN
- EEM Scripts used to Troubleshoot Tunnel Flaps Caused by Invalid Security Parameter Indexes
- Syslog "%CRYPTO-4-RECVD_PKT_MAC_ERR:" Error Message with Ping Loss Over IPsec Tunnel Troubleshooting

Troubleshoot High CPU Utilization

It is usually found that at times the CPU utilization spikes up for a very short period of time and at indeterminate times. Hence it becomes very difficult to run commands that need to be run at the time of the high CPU utilization. This is when an EEM script can be very useful. Set the CPU values at which it should get triggered and it obtains the command outputs.

This is an ***example script*** and should be customized for your requirements:

```
event manager applet capture_cpu_spike
  event snmp oid 1.3.6.1.4.1.9.2.1.56 get-type next entry-op ge entry-val 60
  exit-time 10 poll-interval 1
```

```
action 001 syslog msg "CPU Utilization is high"
action 002 cli command "en"
action 003 cli command "show proc cpu sort | append flash:cpuinfo"
action 004 cli command "show proc cpu sort | append flash:cpuinfo"
action 005 cli command "show stack 236 | append flash:cpuinfo"
action 006 cli command "show call active voice brief | append flash:cpuinfo"
action 007 cli command "show voip rtp connection | append flash:cpuinfo"
action 008 cli command "show isdn call-rate | append flash:cpuinfo"
action 009 cli command "show log | append flash:cpuinfo"
action 010 cli command "show mem stat his | append flash:cpuinfo"
action 011 cli command "show proc cpu his | append flash:cpuinfo"
action 012 cli command "show align | append flash:cpuinfo"
```

The above script not only detects CPU utilization and runs the commands, it also appends the outputs to flash: or any other location of choice. It is triggered when the CPU utilization goes above 60 %. This should be customized to your requirements. However, caution should be exercised to consider the file system free space before you deploy the script.

Related Information

- ***Cisco IOS Network Management Command Reference***
- ***Cisco Support Community − EEM Scripting***
- ***Technical Support & Documentation − Cisco Systems***

Updated: Oct 10, 2013 Document ID: 116176