**Discovery 22: Configuring Cisco IOS Embedded Event Manager (EEM)**

**Introduction**

In this Lab, you use the Embedded Event Manager (EEM) built into IOS to create track and classify events and create notification options for these events.

In this lab you will perform the following activities:

* Use the EEM Syslog event detector to monitor an interface state and apply actions when an event is triggered.
* Use the EEM CLI event detector to monitor CLI events and trigger an action when an event is triggered

What Is EEM?

Embedded Event Manager (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.

Applets and Scripting are two pieces of EEM. Applets are a collection of CLI commands, while Scripts are actions coded in TCL.

Event detectors are used by EEM, and actions provide notifications of the events.

| **Detector** | **Details** |
| --- | --- |
| SNMP | Monitoring SNMP objects. |
| Syslog | Responds to various syslog messages, allowing for matching on regular expressions. |
| Counter | Monitoring and responding to interface counter when cross threshold settings. |
| CLI events | Screening CLI input for a regular expression match. |
| None | This event detector is used to test EEM script/applet using "event manager run" command. |
| Timers | (Countdown, watchdog and CRON) |
| IP SLA and Netflow events |  |

Actions that can be used are:

* Send an email message
* Execute a Cisco command
* Generate an SNMP trap
* Reload the device
* Generate prioritized syslog messages
* Change to a secondary processor in a redundant platform
* Request system information when the event occurs (examples: show tech, show process, CPU history)

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 when the problem begins.

**Task 1: Use the Syslog Event Detector with EEM**

**Activity**

In this task, you will create an EEM script that will monitor a specific interface for a state change to “administratively down.” If the event is triggered the router will perform a command sequence to re-enable the interface and will send an email to an administrator informing them of the occurrence.

**Step 1:** From the router R1 CLI, access configuration mode using the following commands:

R1# enable

R1# configure terminal

R1#



**Step 2:** From configuration mode on the R1 console, create the following EEM script:

event manager applet interface\_Shutdown

event syslog pattern "Interface Ethernet0/0, changed state to administratively down"

action 1.0 cli command "enable"

action 1.5 cli command "config t"

action 2.0 cli command "interface e0/0"

action 2.5 cli command "no shutdown"

action 3.0 cli command "end"

action 3.5 cli command "who"

action 4.0 mail server "192.168.1.1" to ".engineer@cisco.com." from ".EEM@cisco.com." subject ".ISP1\_GigabitEthernet0/0\_SHUT." body "Current users $\_cli\_result"

end

The script you are creating is called “interface\_Shutdown” and it is monitoring syslog messages for the string pattern “Interface Ethernet0/0, changed state to administratively down.” When this pattern is matched, it will trigger the action list. The action list first enters interface configuration mode and issues the no shutdown command, then sends an email to a monitoring email address explaining the action that was observed. The actions will be taken in the order that they are listed in the script.

Enter the following commands:

R1# configure terminal Enter configuration commands, one per line. End with CNTL/Z.

R1(config)# event manager applet interface\_Shutdown

R1(config-applet)# event syslog pattern "Interface Ethernet0/0, changed state to administratively down"

R1(config-applet)# action 1.0 cli command "enable"

R1(config-applet)# action 1.5 cli command "config t"

R1(config-applet)# action 2.0 cli command "interface e0/0"

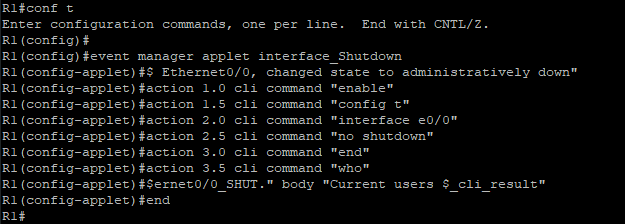
R1(config-applet)# action 2.5 cli command "no shutdown"

R1(config-applet)# action 3.0 cli command "end"

R1(config-applet)# action 3.5 cli command "who"

R1(config-applet)# action 4.0 mail server "192.168.1.1" to ".engineer@cisco.com." from ".EEM@cisco.com." subject ".ISP1\_Interface\_Ethernet0/0\_SHUT." body "Current users $\_cli\_result" R1(config-applet)# end

R1#

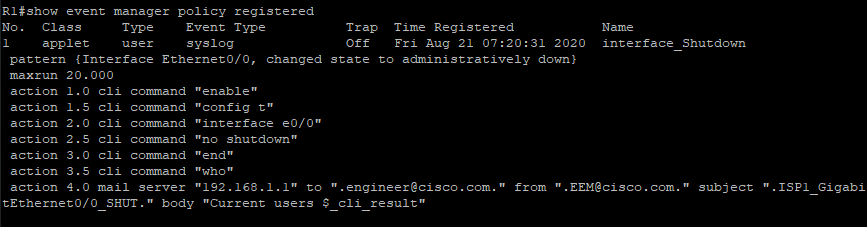


**Step 3:** Verify that the script has been created successfully in the EEM policy, using the following command:

show event manager policy registered

Enter the following command, and observe the results:

R1# show event manager policy registered



**Step 4:** From the R1 console, enable EEM CLI debugging and trigger the EEM event by shutting down the Ethernet 0/0 interface using the following commands:

R1# debug event manager action cli

Debug EEM action cli debugging is on

R1# configure terminal

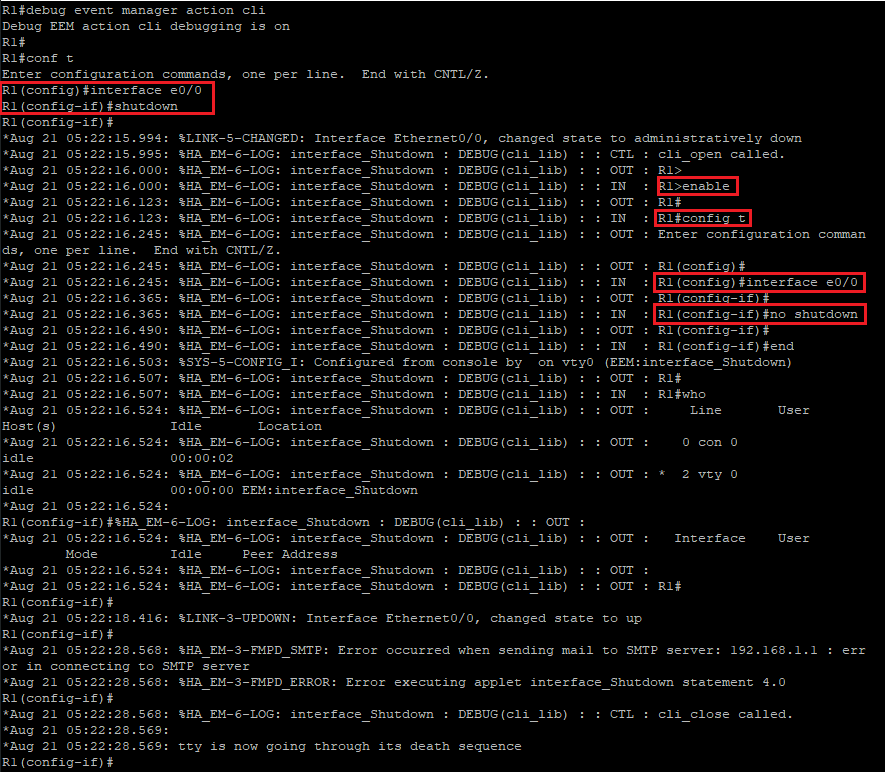
Enter configuration commands, one per line. End with CNTL/Z.

R1(config)# interface e 0/0

R1(config-if)# shutdown

R1(config-if)#

Observe the debug output:

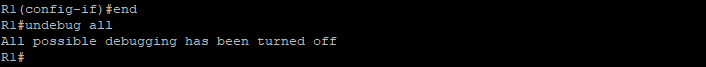


The output that is highlighted above shows that the commands that were listed in the EEM script were carried out in the order that they were listed. The interface state was changed to “up.” You can also observe that there was an error when connecting to the SMTP server, as this lab has limited connectivity. If using email alerts in a production environment you should ensure that emails are being sent to your corporate email server, and that the router has permissions to send email to that server. The lab has no email server configured, so the last lines show that the email has failed to be delivered.

**Step 5:** Disable all debugging on the R1 using the following commands:

R1(config-if)# end

R1# undebug all



**Task 2: Use the CLI Event Detector with EEM**

**Activity**

**Step 1:** From the R1 console access configuration mode using the following command:

R1# configure terminal

R1 (config)#

****

**Step 2:** From configuration mode on the R1 console, create the following EEM script:

event manager applet Stop\_Debug

event cli pattern "debug" sync no skip yes action 1.0 cli command "enable"

action 2.0 cli command "sh users | append unix:Debug"

action 3.0 cli command "sh clock | append unix:Debug\_clock"

end

The script you are creating is called “Stop\_Debug” and it is monitoring the CLI for the debug command. When this pattern is matched, it will skip the command so that it does not take effect. The action list first enters the enabled mode and issues the show users | append flash:Debug command. This command will append the output from the show users command to the end of a file in “Flash” (the lab uses the unix directory as a flash equivalent) called Debug. The next action will then append the current time stamp to the end of the file in “Flash” (the lab uses the unix directory as a flash equivalent) named Debug\_clock. By matching the order of the entries in both files, you will have a list of the users that tried to enter the debug command and the date and time that the user attempted it.

Enter the following information:

R1# configure terminal

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

R1(config)# event manager applet Stop\_Debug

R1(config-applet)# event cli pattern "debug" sync no skip yes

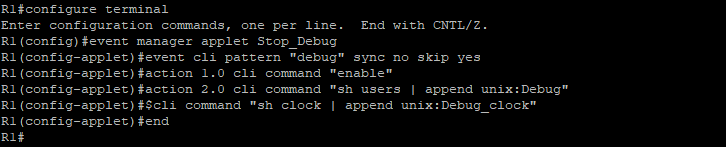
R1(config-applet)# action 1.0 cli command "enable"

R1(config-applet)# action 2.0 cli command "sh users | append unix:Debug"

R1(config-applet)# action 3.0 cli command "sh clock | append unix:Debug\_clock"

R1(config-applet)# end

R1#

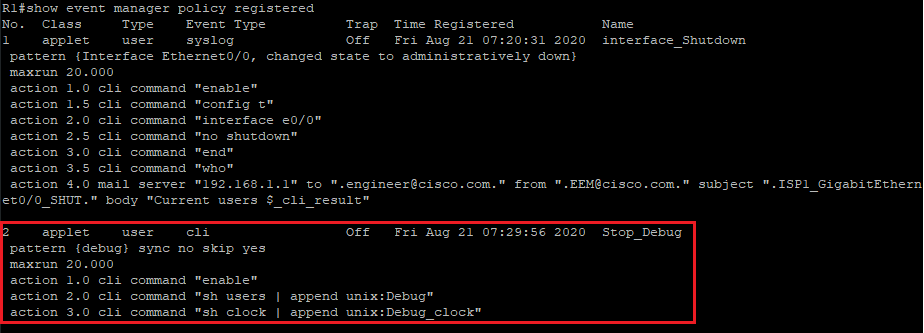


**Step 3:** Verify that the script has been created successfully in the RRM policy using the following command: (Use the Enter key to view the entire output, until you receive the "R1#" prompt)

show event manager policy registered

Enter the following command and observe the output:

R1# show event manager policy registered



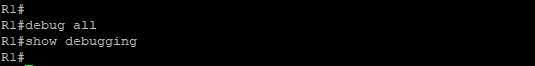
**Step 4:** From the R1 console, trigger the EEM event by attempting to enable the debug all command and verify that the debug all command did not take effect.

Enter the following command:

R1# debug all

R1# show debugging

R1#



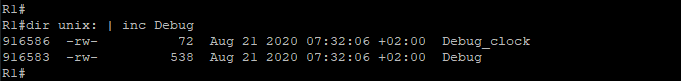
Even though the debug all command appeared to be taken successfully the EEM script had the instruction to “skip” the command. As shown from the show debugging command no debugging has been enabled on the router.

**Step 5:** Verify that the Debug and Debug\_clock files were created in Flash using the following command:

dir unix: | include Debug

Enter the following command and observe the output:

R1# dir unix: | inc Debug

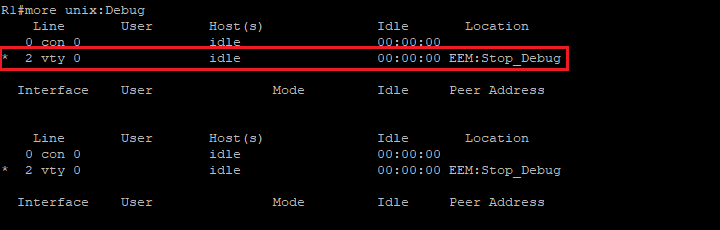


**Step 6:** Inspect the contents of the Debug and Debug\_clock files that were created or appended in Flash using the following command:

more unix:Debug more unix:Debug\_clock

Enter the following command and observe the output:

R1# more unix:Debug



R1#more unix:Debug\_clock



In the output from the first file, you can see the line that was used when the EEM event was triggered. If a user was authenticated to that line their username would also be shown.

In the output from the second file, you can see the date and time that the event was triggered. If more than one event populated the files you would match the first event in one file to the first event in the second file, the second event in the first file to the second event in the second file, and so on.

Summary and Next Steps

EEM can be a very useful tool for the network engineer. An engineer can use EEM for monitoring and correcting many events on Cisco network devices. In this lab, you demonstrated the use of EEM event detectors and associated actions that can be taken.