TOMORROW starts here.
Onboard Automation with Cisco IOS Embedded Event Manager

BRKNMS-2030

Joe Clarke, Distinguished Services Engineer
Session Abstract

Did you ever wish that your routers and switches could become smarter to monitor its own transient error conditions and conduct automated diagnostics? Did you ever struggle with some repetitive tasks that you hope you could have the devices do them automatically? Did you ever dream that your router can automatically configure itself when a new device is plugged in?

Come and join us for the session on an in-depth review of the Cisco IOS Embedded Event Manager (EEM) which allows you to achieve all these things and more that you never thought possible before. This session covers the following topics.

- EEM Architecture
- Understanding Event Detectors
- Understanding EEM Policies
- EEM real world use case review
- Latest EEM feature update
- Introduction to "Embedded Automation Systems (EASy)", a program offering EEM-based solutions in the area of onboard diagnostics, deployment simplification, high availability and more
- EEM new feature and use case demo

This session is for attendees who have worked with Cisco IOS, and who wish to strengthen their skill set to provide improved manageability for their network.
Agenda

- Embedded Event Manager (EEM) Overview
- Understanding EEM Event Detectors
- Understanding EEM Policies
- Advanced EEM Features
- EEM Demos
- EEM Resources
Embedded Event Manager (EEM) Overview
Embedded Event Manager (EEM) Basic Architecture

- Extremely flexible and powerful subsystem within Cisco IOS Software
- Adapt device behavior and insert custom logic without IOS upgrade
- 24 Event Detectors (ED) integrated with IOS modules for wide range of system event detection
- CLI and Tcl based policy provides consistent programmability interface
- Powerful event engine supporting multi-event correlation, advance scheduling and more

Cisco IOS

- Embedded Event Manager Server
  - EEM Applet (CLI-based) Policy
  - EEM Tcl Policy
  - IOS.sh Policy
  - IOS Subsystems
- Event Detectors
  - CLI
  - Syslog
  - Timer
  - Counter
  - OIR
  - Interface
  - Resource Threshold
  - Neighbor Discovery
  - NetFlow
  - Object Tracking
  - XML-RPC
  - RF
  - SNMP
  - SNMP Notification
  - SNMP Object
  - IOS Watchdog
  - MAC Address
- Cisco IOS Infrastructure and Subsystems

BRKNMS-2030 © 2013 Cisco and/or its affiliates. All rights reserved. Cisco Public
Getting an Insider’s View with EEM

- **RELIABLE** — Captures reliable information within the box when connectivity to external systems are not available or reliable

- **QUICK** — Onboard logic provides instant reaction when certain condition is detected and wins precious time to capture critical information

- **DETAILED** — An insider’s view allows you to get more granularity information than you could have afforded through external communication

- **EVENT-DRIVEN** — EEM supports many event detectors integrated with IOS modules to generate event and allow you to avoid constant polling

- **DISTRIBUTED** — Scripts are distributed to each network device and runs locally when triggered supported distributed and collaborative processing for complex network management tasks
What Can EEM Do for Me?

**Challenge 1:** Every few weeks a router is running low on memory around 2 am, and I want to find out what’s happening

• **Solution:** EEM script could be triggered based on the memory utilization, capture the memory information and send the output with Syslog or Email

**Challenge 2:** If my ACL configuration gets changed, I want to get notified, but I can’t sit there monitor it all the time

• **Solution:** EEM script could be triggered by CLI command, take a snapshot of the logged in user, changed configuration, and send an email to you

**Challenge 3:** I want to save energy, but I can’t go around turn off everyone’s IP phone everyday

• **Solution:** Timer ED can be used to trigger the execution of an EEM script to turn off your IP phone at 7pm everyday and turn it back on 7am the next day
## Where Can I Find It?
### -- EEM Version/Product Support Matrix

#### CISCO ACCESS ROUTERS - Current Models

<table>
<thead>
<tr>
<th>EEM Version</th>
<th>Cisco 800 Series</th>
<th>Cisco 1800 Series</th>
<th>Cisco 2800 Series</th>
<th>Cisco 3800 Series</th>
<th>Cisco 1900 Series</th>
<th>Cisco 2900 Series</th>
<th>Cisco 3900 Series</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td></td>
<td>12.3(11)T</td>
<td>12.3(11)T</td>
<td>12.3(11)T</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.1</td>
<td></td>
<td>12.3(14)T1</td>
<td>12.3(14)T1</td>
<td>12.3(14)T1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.1.1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.2</td>
<td>12.4(2)T</td>
<td>12.4(2)T</td>
<td>12.4(2)T</td>
<td>12.4(2)T</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.3</td>
<td>12.4(2)</td>
<td>12.4(11)T</td>
<td>12.4(11)T</td>
<td>12.4(11)T</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.4</td>
<td>12.4(20)T</td>
<td>12.4(20)T</td>
<td>12.4(20)T</td>
<td>12.4(20)T</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3.0</td>
<td>12.4(22)T</td>
<td>12.4(22)T</td>
<td>12.4(22)T</td>
<td>12.4(22)T</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3.1</td>
<td>15.0(1)M</td>
<td>15.0(1)M</td>
<td>15.0(1)M</td>
<td>15.0(1)M</td>
<td>15.0(1)M</td>
<td>15.0(1)M</td>
<td>15.0(1)M</td>
</tr>
<tr>
<td>3.2</td>
<td>15.1(3)T</td>
<td>15.1(3)T</td>
<td>15.1(3)T</td>
<td>15.1(3)T</td>
<td>15.1(3)T</td>
<td>15.1(3)T</td>
<td>15.1(3)T</td>
</tr>
<tr>
<td>4.0</td>
<td></td>
<td>15.2(2)T</td>
<td>15.2(2)T</td>
<td>15.2(2)T</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

#### CISCO ACCESS ROUTERS - Old Models

<table>
<thead>
<tr>
<th>EEM Version</th>
<th>Cisco 1700 Series</th>
<th>Cisco 2600 Series</th>
<th>Cisco 2600XM Series</th>
<th>Cisco 2691 Series</th>
<th>Cisco 3600 Series</th>
<th>Cisco 3700 Series</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td>12.3(4)T</td>
<td>12.3(4)T</td>
<td>12.3(4)T</td>
<td></td>
<td>12.3(4)T</td>
<td></td>
</tr>
<tr>
<td>2.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.1</td>
<td>12.3(14)T1</td>
<td>12.3(14)T1</td>
<td>12.3(14)T1</td>
<td>12.3(14)T1</td>
<td>12.3(14)T1</td>
<td>12.3(14)T1</td>
</tr>
<tr>
<td>2.1.1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.2</td>
<td>12.4(2)T</td>
<td>12.4(2)T</td>
<td>12.4(2)T</td>
<td>12.4(2)T</td>
<td>12.4(2)T</td>
<td>12.4(2)T</td>
</tr>
<tr>
<td>2.4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3.1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3.2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
# EEM Version/Product Support Matrix

## CISCO SERVICE AGGREGATION/CORE ROUTERS

<table>
<thead>
<tr>
<th>EEM Version</th>
<th>Cisco ASR1000 Series</th>
<th>Cisco 7200 Series</th>
<th>Cisco 7301</th>
<th>Cisco 7304</th>
<th>Cisco 7600 Series</th>
<th>Cisco UBR 10000</th>
<th>Cisco UBR 7200</th>
<th>Cisco 12000 Series</th>
<th>Cisco XR 12000</th>
<th>Cisco CRS-1</th>
<th>Cisco ASR 9000</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>12.0(26)S</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>FM</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.1</td>
<td>12.3(14)T1</td>
<td>12.3(14)T1</td>
<td>12.2(28)SB</td>
<td>12.2(18)SXFS</td>
<td>12.2(28)SB</td>
<td>12.2(28)SB</td>
<td></td>
<td>FM</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.2</td>
<td>12.4(2)T</td>
<td>12.4(2)T</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>FM</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.3</td>
<td>2.1XE</td>
<td>12.4(11)T</td>
<td>12.2(33)SB</td>
<td>12.2(33)SB</td>
<td>12.2(33)SB</td>
<td>12.2(33)SB</td>
<td>12.2(33)SB</td>
<td>FM</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.4</td>
<td>12.2(33)XN RL57</td>
<td>12.4(20)T</td>
<td>12.2(33)SRE</td>
<td>12.2(33)SRE</td>
<td>12.2(33)SRE</td>
<td>12.2(33)SRE</td>
<td>12.2(33)SRE</td>
<td>FM</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3.0</td>
<td>12.2(33)XN RL57</td>
<td>12.4(22)T</td>
<td>12.2(33)SRE</td>
<td>12.2(33)SRE</td>
<td>12.2(33)SRE</td>
<td>12.2(33)SRE</td>
<td>12.2(33)SRE</td>
<td>FM</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3.1</td>
<td>15.1(3)S3</td>
<td>15.0(1)M</td>
<td>Planning</td>
<td>Planning</td>
<td>Planning</td>
<td>Planning</td>
<td>Planning</td>
<td>Planning</td>
<td>Planning</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3.2</td>
<td>Planning</td>
<td>15.1(3)T</td>
<td>Planning</td>
<td>Planning</td>
<td>Planning</td>
<td>Planning</td>
<td>Planning</td>
<td>Planning</td>
<td>Planning</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4.0</td>
<td>Planning</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Planning</td>
<td>Planning</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

## CISCO CATALYST SWITCHES

<table>
<thead>
<tr>
<th>EEM Version</th>
<th>Catalyst 3000 Switches</th>
<th>Cisco 3400ME Switches</th>
<th>Catalyst 4500 Switches</th>
<th>Catalyst 4900 Switches</th>
<th>Catalyst 6500 Switches</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>12.2(18)SXFS</td>
</tr>
<tr>
<td>2.1.5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>w/ Modularity 12.2(18)SXFS</td>
</tr>
<tr>
<td>2.3</td>
<td>12.2(40)SE</td>
<td>12.2(40)SE</td>
<td>12.2(44)SG</td>
<td>12.2(44)SG</td>
<td>12.2(33)SXH</td>
</tr>
<tr>
<td>2.4</td>
<td>12.2(50)SE</td>
<td>12.2(50)SE</td>
<td>12.2(52)SG</td>
<td>12.2(52)SG</td>
<td>12.2(33)SXI</td>
</tr>
<tr>
<td>3.0</td>
<td>12.2(52)SE</td>
<td>12.2(52)SE</td>
<td>12.2(54)SG(1)</td>
<td>12.2(54)SG(1)</td>
<td></td>
</tr>
<tr>
<td>3.1</td>
<td>12.2(52)SE</td>
<td>12.2(52)SE</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3.2</td>
<td>12.2(52)SE</td>
<td>12.2(52)SE</td>
<td>15.0(2)SG</td>
<td>15.0(2)SG</td>
<td></td>
</tr>
<tr>
<td>4.0</td>
<td>Planning</td>
<td>Planning</td>
<td></td>
<td></td>
<td>15.1(1)SY</td>
</tr>
</tbody>
</table>

*Shipping, Coming Soon, Planning*
<table>
<thead>
<tr>
<th>Event Detector</th>
<th>Description</th>
<th>EEM Version in IOS</th>
<th>IOS XR</th>
<th>NX-OS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Syslog</td>
<td>RegExp match of local syslog message</td>
<td>✓</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>SNMP Notif</td>
<td>SNMP MIB Variable Threshold</td>
<td>✓</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>Watchdog</td>
<td>IOS process or subsystem activity events</td>
<td>✓</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>Interface Counter</td>
<td>(Interface) Counter Threshold</td>
<td>✓</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>Timer</td>
<td>Designated Time or Interval</td>
<td>✓</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>Counter</td>
<td>Change of a designated counter value</td>
<td>✓</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>Application specific</td>
<td>An IOS subsystem or policy script</td>
<td>✓</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>CLI</td>
<td>RegExp match of input via command line interface</td>
<td>✓</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>OIR</td>
<td>Hardware online insertion and removal OIR</td>
<td>✓</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>none</td>
<td>No trigger, used in conjunction with exec command</td>
<td>✓</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>ERM</td>
<td>Embedded Resource Manager (ERM) events</td>
<td>✓</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>EOT</td>
<td>Enhanced Object Tracking variable (EOT) events</td>
<td>✓</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>RF</td>
<td>IOS Redundancy Facility (switchover)</td>
<td>✓</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>GOLD</td>
<td>Generic Online Diagnostics (GOLD) events</td>
<td>✓</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>SNMP Proxy</td>
<td>Incoming remote SNMP Notification</td>
<td>✓</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>XML RPC</td>
<td>Incoming XML message</td>
<td>✓</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>Routing</td>
<td>State change of Routing Protocols</td>
<td>✓</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>Netflow</td>
<td>Traffic Flow information from Netflow</td>
<td>✓</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>IPSLA</td>
<td>IPSLA events (supersedes EOT for EEM / IPSLA)</td>
<td>✓</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>CLI enhanced</td>
<td>Integrates CLI Ed with the XML PI</td>
<td>✓</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>SNMP Object</td>
<td>Intercept SNMP GET/SET requests</td>
<td>✓</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>Neighbor Disco</td>
<td>CDP, LLPD, Link up/down events</td>
<td>✓</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>Identity</td>
<td>802.1x and MAB authentication events</td>
<td>✓</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>MAC</td>
<td>MAC Address Table entry changes</td>
<td>✓</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>Enhanced CLI ED</td>
<td>Provide AAA-like info to CLI policies</td>
<td>✓</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>ESM enhancements for syslog ED</td>
<td>Filter syslog messages in line</td>
<td>✓</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>Hardware</td>
<td>Register for environmenta monitoring hardware</td>
<td>✓</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>Statistics</td>
<td>Threshold crossing of a statistical counter</td>
<td>✓</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>Fan (absent / bad)</td>
<td>Presence and State of a Fan</td>
<td>✓</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>Module failure</td>
<td>Occurrence of a Module Failure Event</td>
<td>✓</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>Storm Control</td>
<td>Occurrence of a Storm Control Event</td>
<td>✓</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>Temperature</td>
<td>Temperature Sensor Thresholds</td>
<td>✓</td>
<td></td>
<td>✓</td>
</tr>
</tbody>
</table>
Understanding EEM Event Detectors
Event Detection and Filtering

Event manager applet email_hsrp_state_change

Event syslog pattern "\*\*%HSRP-5-STATECHANGE.\*"

action 1.0 info type routename
  action 1.1 cli command "enable"
  action 1.2 cli command "del /force flash:hsrp_state_change"
  action 1.3 cli command "show standby | append hsrp_state_change"
  action 1.4 cli command "show standby brief | append hsrp_state_change"
  action 1.5 cli command "show tech | append hsrp_state_change"
  ...
EEM Event Detectors

- **EEM 1.0**
  - SNMP ED (IOS, IOS-XR, NX-OS)
  - Syslog ED (IOS, IOS-XR, NX-OS)

- **EEM 2.0**
  - Application ED (IOS, IOS-XR)
  - Interface ED (IOS)
  - Counter ED (IOS, IOS-XR, NX-OS)
  - Timer ED (IOS, IOS-XR)
  - Watchdog ED (IOS, IOS-XR)

- **EEM 2.1**
  - CLI ED (IOS, IOS-XR, NX-OS)
  - None ED (IOS, IOS-XR)
  - Online Insertion Removal (OIR) ED (IOS, IOS-XR, NX-OS)
  - GOLD ED (IOS)

- **EEM 2.2**
  - Embedded Object Tracking (EOT) ED (IOS, IOS-XR, NX-OS)
  - Resource ED (IOS)
  - Redundancy framework ED (IOS)

- **EEM 2.4**
  - SNMP notification ED (IOS, IOS-XR)
  - XML-RPC ED (IOS)

- **EEM 3.0**
  - Routing ED (IOS)
  - NetFlow ED (IOS)
  - IP SLA ED (IOS)

- **EEM 3.1**
  - SNMP Object ED (IOS)

- **EEM 3.2**
  - Neighbor Discovery ED (IOS)
  - Identity ED (IOS)
  - MAT ED (IOS)

- **EEM 4.0**
  - More parameters passed to CLI ED (IOS)

- **Un-versioned Event Detectors on IOS-XR and NX-OS**
  - Hardware ED (IOS-XR, NX-OS)
  - Process ED (IOS-XR)
  - Storm-control ED (NX-OS)
  - Sysmgr ED (NX-OS)
## EEM Event Detector – Syslog ED

### Functionality
Triggers Event on Matches for Syslog Messages Based on Regular Expression

### Example
*event syslog pattern ".*UPDOWN.*Ethernet1/0.*" occurs 4*

### Use Case
Troubleshooting, Automatic Fault Detection and Alert
## EEM Event Detector – SNMP ED

### Functionality

Triggers Event Based on SNMP OID Value Crossing Predefined Threshold

### Example

```
event snmp oid 1.3.6.1.4.1.9.9.109.1.1.1.1.3.1 get-type exact entry-op ge entry-val 75 poll-interval 10
```

### Use Case

System Stats Monitoring and Alerting, e.g. CPU and Memory Utilization
EEM Event Detector – Interface ED

Functionality

Triggers Event When Interface Counters Cross Threshold. 22 Counters Supported, Including `input_error`, `interface_reset`, `transmit_rate`, etc.

Example

```
event interface name FastEthernet0/0 parameter receive_throttle entry-op ge entry-val 5 
  entry-type increment poll-interval 90
```

Use Case

Real Time Alert and Recovery of Interface Error
EEM Event Detector – Timer ED

Functionality

Triggers Events on Watchdog, Count Down, cron and Absolute Timer

Example

event timer cron cron-entry "0 19 * * 0-7"
event timer watchdog time 30

Use Case

System Monitoring via Periodic Action, Periodic Data Collection and Reporting


## EEM Event Detector – CLI ED

### Functionality

Triggers Synchronous or Asynchronous Events When Certain CLI Is Executed. Allow Custom CLI Creation (EEM 3.0).

### Example

```
event cli pattern "write mem.*" sync yes
```

### Use Case

Config Management, Security, Feature Customization
## Functionality

Triggers Events When Incoming or Outgoing (EEM 3.1) Trap Is Intercepted

### Example

| event `snmp-notification` oid 1.9.9.9.1 oid-val 1 op eq src-ip-address ADDR direction incoming |

### Use Case

Trap Logging, Inter-Box Communication
EEM Event Detector – XML-RPC ED

Functionality

Triggers Events When a Specifically Formatted SOAP Request Is Received

Example

```xml
<SOAP:Body>
  <run_eemscript>
    <script_name> name of script </script_name>
    <argc> argc value </argc>
    <arglist> argv1 value </l> ... <l> argvn value </l> </arglist>
  </run_eemscript>
</SOAP:Body>

<SOAP:Body>
  <run_eemscript_response>
    <return_code> rc </return_code>
    <output> output string </output>
  </run_eemscript_response>
</SOAP:Body>
```

Use Case

Network Management -- Custom Communication with External Server and NMS Station
EEM Event Detector – Netflow ED

**Functionality**

Detects Events Related to Flexible NetFlow

**Example**

Router(config)# flow monitor my-monitor-1
Router(config-flow-monitor)# record netflow-original
Router(config)# interface ethernet 0/0
Router(config-if)# ip flow monitor my-monitor-1 input
Router(config)# event manager applet netflow
Router(config)# event nf monitor-name "my-monitor-1" event-type create exit-event-type
delete event1 entry-value "192.168.1.1"
extit-value "192.168.1.1"
field ipv4 destination address entry-op eq exit-op eq
Router(config)# action msg syslog msg "netflow applet triggered"

**Use Case**

Security, Network Monitoring
EEM Event Detector – IPSLA ED

Functionality

Trigger Events When IPSLA Test Results Cross Certain Threshold. Integrated with Auto IPSLA Group to Monitor Large Number of IPSLA Operation Results

Example

event manager applet watch-jitter
  event ipsla operation-id 1 reaction-type jitterAvg
  action 001 cli command "enable"
  action 002 if $_ipsla_measured_threshold_value > $_ipsla_threshold_rising
  action 003 cli command "config t"
  action 004 cli command "ip route 10.10.20.0 255.255.255.0 192.168.15.1"
  action 005 cli command "end"
  ...

Use Case

Link Failure Detection, Diagnostics and Recovery
EEM Event Detector – Neighbor Discovery ED

Functionality

Triggers Events When CDP or LLDP Message Is Detected and Reports All CDP and LLDP Fields Associated with the Events

Example

event manager applet cdp-phone
  event neighbor-discovery interface .* cdp add
  action 1.0 if $nd_cdp_capabilities_bit_4 eq "YES"
  action 1.1 if $nd_cdp_platform eq "Cisco IP Phone"
  action 1.2 cli command "conf t"
  action 1.3 cli command "interface gi1/0/2"
  action 1.4 cli command "macro description CISCO_PHONE_AUTO_SMARTPORT"
  action 1.5 cli command "switchport access vlan 1"

Use Case

Device Auto Configuration
Understanding EEM Policies
EEM Applet (CLI-Based) Policy

- Supported on IOS and NX-OS
- EEM CLI-based policy made of three parts:
  - Applet name
  - Event filtering spec
  - List of things to do (action)
- Stored as part of configuration
- Support programming control structure as of EEM 3.0 (IOS only)
- File manipulation supported in EEM 4.0 (IOS only)
- Managed and deployed using standard configuration management tool
EEM Applet Policy Configuration (IOS)

- Step one—configure Env Var
  - Configure necessary EEM environment variable
- Step two—register policy
  - Register the policy with the applet policy engine
- Step three—establish event trigger
  - Define the event that triggers the policy
- Step four—define action
  - Define the action to take upon detection of the event

**event manager environment**
```plaintext
_email_server 172.27.121.177
```

**event manager applet**
```plaintext
event_hsrp_state_change
```

**event syslog**
```plaintext
pattern ".*.HSRP-5-STATECHANGE.*"
```

**action 1.0 info**
```plaintext
type routername
```

**action 1.1 cli**
```plaintext
command "enable"
```

**action 1.3 cli**
```plaintext
command "show standby | append hsrp_state_change"
```

**action 1.4 cli**
```plaintext
command "show standby brief | append hsrp_state_change"
```

**action 1.8 mail**
```plaintext
server "$_email_server" to "$_email_to"
from "$_email_from" subject "HSRP_STATE_CHANGE Alert from $_info_routername: $_syslog_msg" body "$_cli_result"
```
EEM Applet Policy Example (IOS)
-- Timed IP Phone Turn On/Off

```plaintext
event manager applet phone_off
  event timer cron cron-entry "0 19 * * 0-7"
  action 1 cli command "enable"
  action 2 cli command "config term"
  action 3 cli command "interface GigabitEthernet0/10"
  action 4 cli command "shutdown"
  action 5 syslog msg "ip phone OFF"

event manager applet phone_on
  event timer cron cron-entry "0 7 * * 0-7"
  action 1 cli command "enable"
  action 2 cli command "config term"
  action 3 cli command "interface GigabitEthernet0/10"
  action 4 cli command "no shutdown"
  action 5 syslog msg "ip phone ON"
```

Time ED Set to Trigger Policy 19:00

Shut Down Interface to Turn Off IP Phone

Time ED Set to Trigger Policy 7:00

Turn on Interface to Turn On IP Phone
Control Structure Support for Applet Policies (IOS)

- Interactive CLI support allows applet to read and write on active terminal
- EEM 3.0 introduced control structure for EEM applet policy, e.g., if-else and for each

**Example:** Generate Error Message When Email Server Not Reachable

```plaintext
event manager applet example1
event none
action 100 handle-error type ignore
action 101 mail server "noserver" to "jclarke@cisco.com" from "jclarke@cisco.com" subject "hi" body "hi"
action 102 set mail_error $_error
action 103 if $mail_error ne FH_EOK
action 104  if $mail_error eq FH_ESMTPCONNECT
action 105  puts "Error connecting to mail server: action $_error_label"
action 106  else
action 107  puts "Error: $_error sending mail: action $_error_label"
action 108  end
action 109 end
```
Applet-Based File Manipulation (IOS)

- EEM 4.0 introduced the ability to read and write files from applets
- Operations exist for reading, writing, appending, and deleting files on flash and from remote locations (where supported)

**Example:** Transfer a file from flash to a TFTP server

```
event manager applet file-example
  event none
  action 100 file open in flash:
data.txt r
  action 200 file open out tftp://10.1.1.1/data.txt w
  action 300 file read in contents
  action 400 file puts out $contents
  action 500 file close in
  action 600 file close out
```

- Open the input file for reading
- Open the output file file on the TFTP server for writing
- Read all of the contents from the input file
- Write all of the contents to the remote file
EEM Applet Policy Configuration (NX-OS)

- **Step one—configure Env Var**
  - Configure necessary EEM environment variable

- **Step two—register policy**
  - Register the policy with the applet policy engine

- **Step three—establish event trigger**
  - Define the event that triggers the policy

- **Step four—define action**
  - Define the action to take upon detection of the event

- **Step one—configure Env Var**
  - event manager environment
    
    my_interface “Ethernet1/4”

- **Step two—register policy**
  - event manager applet cfg_interface

- **Step three—establish event trigger**
  - event syslog pattern "ETHPORT-5-IF_DOWN: Interface Ethernet1/3"

- **Step four—define action**
  - action 1.0 cli config t
  - action 2.0 cli int $my_interface
  - action 3.0 cli no shut
  - action 4.0 cli end
EEM Tcl-Based Policy

- Supported on IOS and IOS-XR (same syntax)
- Provide programmable interface using Tcl 8.3.4 with Cisco EEM extension
- Safe-Tcl supported to provide sandbox for Tcl-based policies and protect hosting environment
- Fast Tcl policy support introduced in EEM 3.0 minimize the initialization overhead for frequent running Tcl policies (IOS)
- Tcl byte-code support for code obfuscation (IOS)
- Digital signature supported (EEM 3.0) for trusted policy validation (IOS)
EEM Tcl-Based Policy

- **Step one**—register user directories
  - Register user policy directory and user library directory

- **Step two**—code policies offline
  - No online editor available

- **Step three**—download policy
  - Download Tcl policies using standard IOS file transfer mechanisms
  - Support script auto refresh from remote location

- **Step four**—EEM environment variable configuration

- **Step five**—register policy
  - Register policy to Tcl policy engine

---

```bash
mkdir disk2:/eem
```

**event manager directory user policy** disk2:/eem

**event manager directory user library** disk2:/eemlib

```bash
copy tftp disk2:/eem
Address or name of remote host []? 2.2.2.2
Source filename []? sl_cfgSaveRemT.tcl
Destination filename [sl_cfgSaveRemT.tcl]? eem/sl_cfgSaveRemT.tcl
Accessing tftp://2.2.2.2/sl_cfgSaveRemT.tcl...!
1232 bytes copied in 0.620 secs (1987 bytes/sec)
```

**event manager update user policy** group “*.tcl”

**repository** tftp://2.2.2.2/users2/jclarke/eem_1

```bash
event manager environment _email_server
172.27.121.177
```

**event manager policy** sl_cfgSaveRemT.tcl type user
Example: EEM Tcl-Based Policy
Suspend Unused Ports

::cisco::eem::event_register_syslog pattern "LINEPROTO-5-UPDOWN" maxrun 600

if { ![info exists suspend_ports_config] } {
    set result "ERROR: Policy cannot be run: variable suspend_ports_config has not been set"
    error $result $errorInfo
}

namespace import ::cisco::eem::*
namespace import ::cisco::lib::*

proc run_cli {clist} { … }

array set arr_einfo [event_reqinfo]
if { ![regexp {Interface ([^,]+), changed state to up} $arr_einfo(msg) -> iface] } {
    exit
}

while { 1 } {
    set results [run_cli [list "show event manager policy pending | include tm_suspend_ports.tcl"]]
    if { ![regexp {tm_suspend_ports.tcl} $results] } {
        break
    }
    after 1000
    if { [catch {open $suspend_ports_config "r"} result] } {
        exit
    }
    …

Say you want to suspend (i.e., shutdown) ports that haven’t been active in a week…

…Use EEM to watch for ports that become active…
Example: EEM Tcl-Based Policy
Suspend Unused Ports (Cont.)

```
set fd $result
set contents [read $fd]
close $fd

set contents [string trim $contents]
array set ports [split $contents]

if { [info exists ports($iface)] } {
    array unset ports $iface

    set fd [open $suspend_ports_config "w"]
    puts -nonewline $fd [array get ports]
    close $fd
}

...Then delete those newly active ports from a log file tracking ports that are down.
```
Every night at midnight, another EEM policy runs that records each “down” port into a file…
Example: EEM Tcl-Based Policy
Track Ports That Are Down (Cont.)

```tcl
set result [run_cli [list "show ip interface brief | include Ethernet"]]
foreach line [split $result "]n"] {
    set line [string trim $line]
    regsub -all \$+ $line "\" line
    set elems [split $line]
    set iface [lindex $elems 0]
    if { ![regexp (Ethernet) $iface] || [llength $elems] < 6 } {
        continue
    } elseif {[regexp "administratively" $elems 4] && [regexp "down" $elems 5]} {
        set status $ADMIN_DOWN
    } elseif {[regexp "up" $elems 4] && [regexp "up" $elems 5]} {
        set status $UP
    } else {
        set status $DOWN
    }
    if { [info exists ports($iface)] } {
        if { $status == $UP || $status == $ADMIN_DOWN } {
            array unset ports $iface
        } elseif { [expr $now - $ports($iface)] > $suspend_time } {
            set suspend_ports($iface) $ports($iface)
        }
    } else {
        if { $status == $DOWN } {
            set ports($iface) $now
        }
    }
}
```

...If the port has been down long enough, it is put on a “to-be-shutdown” list.
Example: EEM Tcl-Based Policy
Track Ports That Are Down (Cont.)

```
set fd [open $suspend_ports_config "w"]
puts -nonewline $fd [array get ports]
close $fd

set cli [list "config t"]
foreach port [array name suspend_ports] {
    if { [info exists suspend_quarantine_vlan] } {
        set cli [concat $cli [list "interface $port" "switchport access vlan $suspend_quarantine_vlan"]]
        action_syslog msg "Moving port $port into quarantine VLAN $suspend_quarantine_vlan since it was last used on [clock format $suspend_ports($port)]"
    } else {
        set cli [concat $cli [list "interface $port" "shut"]]
        action_syslog msg "Shutting down port $port since it was last used on [clock format $suspend_ports($port)]"
    }
}
_lappend cli "end"
if { [catch {run_cli $cli} result] } {
    action_syslog priority err msg "Failed to shutdown ports: '$result'"
}
```

For each port in the list, it is administratively shut down or moved to a quarantine VLAN (if said VLAN is defined).

Download the full version from https://supportforums.cisco.com/docs/DOC-39192
Fast Tcl Policy Support

- High-performance Tcl policies and active-event-handling
- Dramatic increase in the policy triggering speed
- New Tcl commands introduced: `event_completion`, `event_wait`, `event_completion_with_wait`

**Example:**

```tcl
::cisco::eem::event_register_none maxrun 0 sync yes
namespace import ::cisco::eem::*
namespace import ::cisco::lib::*
set jpi 1
while {1 == 1} {
    array set arrߩ einfo [event_reqinfo]
    if {$_errno != 0} {
        set result [format "component=%s; subsys err=%s; posix err=%s\n%s\n"
            $_errno_sub_num $_errno_sub_err $_errno_posix_err $_errno_str]
        error $result
    }
    action_syslog priority info msg "fast instance: $jpi"
    puts "fast instance $jpi"
    if {$jpi == 10} {
        exit
    }
    event_completion status 0
    incr jpi
    array set _event_state_arr [event_wait]
    if {$_event_state_arr(event_state) != 0} {
        #action_syslog msg "Exiting with failed event_state" priority info
        puts "Exiting with failed event_state $_event_state_arr(event_state)"
        exit
    }
}
```

Notify EEM Server That the Policy Is Done Servicing Current Event

Put Tcl Policy Into Sleep State
Safe-Tcl

- A safety mechanism that allows untrusted Tcl scripts to run in an interpreter that was created in a safe mode.

- System policies run in full mode and user policies run in safe mode.

- Restricted set of commands prevent user policies from accessing system critical resources. E.g., file access in protected directory, and tighter controls on CLI commands.

- Policies signed with a valid Cisco signature automatically promoted to system policy running in full mode.
EEM IOS.sh-Based Policy

- Supported on IOS only
- A simpler way to define EEM policies
- Fully integrated EEM policy engine with access to all event detectors and event meta data
- IOS.sh-based EEM policies stored as separate files with .sh extension on disk
- Scripting language built into IOS parser with syntax similar to Linux bash shell
- Supports shell variables, control structures, functions, triggers, etc.
- Available in EEM 3.2 and higher
EEM IOS.sh-Based Policy

- **Step one**—register user directories
  - Register user policy directory and user library directory

- **Step two**—code policies offline
  - No online editor available

- **Step three**—download policy
  - Download IOS.sh policies using standard IOS file transfer mechanisms
  - Support script auto refresh from remote location

- **Step four**—EEM environment variable configuration

- **Step five**—register policy
  - Register policy to EEM policy engine

```bash
mkdir disk2:/eem
event manager directory user policy disk2:/eem

copy tftp disk2:/eem
Address or name of remote host []? 2.2.2.2
Source filename []? sl_cfgIntf.sh
Destination filename [sl_cfgIntf.sh]? eem/sl_cfgSaveRemT.sh
Accessing tftp://2.2.2.2/sl_cfgIntf.sh...!
1232 bytes copied in 0.620 secs (1987 bytes/sec)
event manager update user policy group “*.sh”
repository tftp://2.2.2.2/users2/jclarke/eem_1

event manager environment ptrIntf FastEthernet0/1

event manager policy sl_cfgIntf.sh type user
```
IOS.sh-Based Policy Config Example

```bash
##::cisco::eem::event_register_neighbor_discovery interface .* lldp update

fetch CURRENT_TRIGGER /config/interface($INTERFACE)/macro/description
# Predefine the trigger in case no capabilities match
DEVICE_TYPE="Default device"
NEW_TRIGGER=CISCO_LLDPDEVICE_EVENT
if [[ $_nd_lldp_capabilities_bit_7 -eq YES ]]; then
    DEVICE_TYPE="Host"
    NEW_TRIGGER=CISCO_HOST_EVENT
fi
if [[ $_nd_lldp_capabilities_bit_5 -eq YES ]]; then
    DEVICE_TYPE="Phone"
    NEW_TRIGGER=CISCO_PHONE_EVENT
fi
if [[ $_nd_lldp_capabilities_bit_3 -eq YES ]]; then
    DEVICE_TYPE="Access Point"
    NEW_TRIGGER=CISCO_AP_AUTO_SMARTPORT
fi
if [[ $_nd_lldp_capabilities_bit_4 -eq YES ]]; then
    DEVICE_TYPE="Router"
    NEW_TRIGGER=CISCO_ROUTER_EVENT
fi
……..
```

- **Register for the Event to Listen To**
- **Fetch Configuration Data from Parser Database**
- **Check ED Metadata Field to Determine Device Capabilities**
EEM Policy Deployment Options

CLI-Based Policy Deployment
- EEM policy shows up as node configuration
- Easily configured with any configuration management tool

Tcl-Based Policy Deployment with LMS
- LMS 3.1 introduced native support for EEM policy deployment
- Enables quick deployment of EEM policies onto large number of devices

Policy Update from Remote Server
- “event manager update user policy” command automatically updates one or more Tcl policies from remote server
- Allow EEM policies to be stored and maintained at a central location of user’s choice
Advanced IOS EEM Features
Multi-Event Correlation Support (EEM 2.4)

- Support up to 6 events for correlation
- Correlation statement supports “and” and “or” operation
- Frequency based correlation
- Supported for both Tcl and CLI-based EEM policies

Example:

```plaintext
event manager applet example
event tag e1 syslog pattern ".*UPDOWN.*Ethernet1/0.*"
event tag e2 syslog pattern ".*UPDOWN.*Ethernet1/1.*"
event tag e3 syslog pattern ".*UPDOWN.*FastEthernet0/0.*"
trigger occurs 1
correlate event e1 or event e2 or event e3
attribute tag e1 occurs 1
attribute tag e2 occurs 1
attribute tag e3 occurs 1
action 1.0 reload
set 2.0 _exit_status 1
```
Class-Based Policy Scheduling (EEM 3.0)

- Up to 26 non-default policy classes can be configured
- Configurable number of policy execution threads for each class
- New EEM scheduler commands introduced to customize policy scheduling

Example:
- Declare a non-default class in policy registration
  - event manager applet one class A
- Create two EEM execution threads to run applets of the default class
  - event manager scheduler applet thread class default number 2
- Creates one EEM execution thread to run Tcl scripts of class A, B, D, and E
  - event manager scheduler script thread class A B range D-E number 1
- Display detailed information on event manager schedule
  - show event manager thread [queue-type <queue-type>][detailed]
Advanced Deployment Options (EEM 2.4 and 4.0)

- Since EEM 2.4 policies can be copied from a remote server and registered in one shot
  
  Router#event manager update user policy name sl_config_monitor.tcl repository tftp://fdcd:0216:2fd9:e250::33

  - Can be done with new and existing policies
  - Repository can be any URI supported by IOS

- EEM 4.0 introduces remote policy support where policies can reside on a network server and get re-synced each time the device boots

  Router(config)#event manager policy sl_config_monitor.tcl remote tftp://fdcd:0216:2fd9:e250::33

  - DANGER! Security is a concern to make sure malicious code is not loaded when the device reboots
  - Use the new digest feature to specify the MD5 or SHA-1 checksum of your remote policies

  Router(config)#event manager policy sl_config_monitor.tcl remote tftp://fdcd:0216:2fd9:e250::33 checksum md5 2da800f9bb5d158f15f3d077be05e1af
EEM Demos
1. Print interface broadcasts to the local log buffer and to SYSLOG every minute
The Problem

- We would like to monitor how many broadcasts are received on GigabitEthernet0/0
- We want to be kept apprised when the number of broadcasts increases
The Solution

- A “show int gi0/0 | inc broadcasts” command will be run every minute
- The output of that command will be recorded locally to the logging buffer
- …And also sent to SYSLOG for archival and correlation

- We’ll use an applet with a cron timer event detector
2. Same thing, but this time we’ll do it every 5 seconds instead of 1 minute
The Problem

- We just realized that every minute was not fast enough, and we’d like to collect the same information every 5 seconds.

- Cron is ideal to run a policy at a specific time (e.g., on the hour, every Tuesday, on the first of the month, etc.), but when we need more granular intervals, we need something else.
The Solution

- The cron timer will be replaced with the *watchdog* timer
- The watchdog timer will execute a policy periodically based on a specified number of seconds
3. Detect a sudden increase of broadcast packets
The Problem

- Too many messages to look at… 😊
- We want to manage by exception

- Rather than logging the number of broadcast packets, have the router monitor itself and only notify us if there is a sudden spike
The Solution

- The *Interface* Event Detector will be used to track the number of broadcast packets
- One can track an absolute value, average, or variation (interval)
- We’ll trigger a message when there is more than 5 broadcast packets in a 10 second interval
Highlights

- This is the event detector line

```plaintext
event interface GigabitEthernet0/0
  parameter receive_broadcasts entry-op gt entry-val 5 entry-type increment
  poll-interval 10
```

- The interface name where the condition was triggered is stored in the `$interface_name` variable

- Use the `show event manager detector DETECTOR detail` command to see all available variables
4. We want to be notified when the routing table changes
The Problem

- Routes are silently added and removed from the routing table
- Our network should be stable, and any change to the table is an exception
- We want to be notified via SYSLOG if the routing table changes
The Solution

- Use the *Routing* Event Detector to watch the routing table
- Look for any change to any network
- Send the details of the update using the `syslog` action
- **NOTE:** this policy requires EEM 3.0 or higher
Highlights

- This is the event detector line

```
event routing network 0.0.0.0/0 type all ge 1
```

- The network “0.0.0.0/0” combined with the “ge 1” argument will match every potential route
- We can match on specific protocol as well, but this example matches all protocols
5. Create your own custom CLI
The Problem

- The output of the `show environment` command includes temperatures in degrees Celsius.

- American users would prefer to see temperatures in degrees Fahrenheit.
The Solution

- Use an EEM applet to modify the output of the `show environment` command and convert the temperature from degrees Celsius to degrees Fahrenheit.
- Now we only have to remember the conversion formula once!
EEM cannot intercept CLI commands it runs

This command will not be intercepted:

- action 1.0 cli command "show env all"

The applet will print every line of the “show env all” output unchanged except for the line that contains the temperature

```
action 1.9 foreach line "$_cli_result" "\n"
action 2.0 regexp "[0-9]+ Celsius (.*)\r" "$line" ignore state
action 2.1 if $_regexp_result eq 1
action 2.2 puts " System Temperature: $_result Fahrenheit $state"
action 2.3 else
action 2.4 puts "$line"
```
The Code

```
event manager applet show_env
  event cli pattern "^show environment all" sync yes
  action 0.5  cli command "enable"
  action 1.0  cli command "show env all"
  action 1.1  regexp "([0-9]+) Celsius" "$_cli_result" ignore deg
  action 1.2  if $_regexp_result eq 1
    action 1.3  multiply $deg 2
    action 1.4  add $_result 32
    action 1.5  set degf "$_result"
    action 1.6  divide $deg 10
    action 1.7  multiply $_result 2
    action 1.8  subtract $degf $_result
  action 1.9  foreach line "$_cli_result" "\n"
    action 2.0  regexp "[0-9]+ Celsius (.*)\r" "$line" ignore state
    action 2.1  if $_regexp_result eq 1
      action 2.2  puts " System Temperature: $_result Fahrenheit $state"
      action 2.3  else
      action 2.4  puts "$line"
      action 2.5  end
    action 2.6  end
  action 2.7  end
```
6. Automatically fail over to a redundant link when a network path becomes unavailable
The Problem

- Branch office connects to corporate headquarters through a WAN link
- If the path to the headquarters goes down, the redundant VPN tunnel needs to be established
The Solution

- An IP SLA collector will be used to test the connectivity to a node in the headquarters
- If the ping fails, then the backup VPN tunnel will be brought up
- When the IP SLA collector can once again get a response over the main WAN interface, then the tunnel will be brought back down
- We’ll use EEM applets to track the status of the IP SLA collector
When Everything Works…
Routing Goes Through the Main WAN
When Connectivity Is Lost…

Traffic Goes Across the VPN

Remote Office

Backup path

HQ
IP SLA and Track Configuration

```
ip sla 1
  icmp-echo 172.18.123.33 source-ip 14.32.100.90
ip sla schedule 1 life forever start-time now
track 1 ip sla 1 reachability
delay up 60
```
event manager applet wan_down
  event track 1 state down
  action 1.0 syslog msg "WAN network is no longer passing traffic"
  action 2.0 cli command "enable"
  action 3.0 cli command "config t"
  action 4.0 cli command "int tun0"
  action 5.0 cli command "no shut"
  action 6.0 cli command "end"
  action 7.0 syslog msg "Failed over to tunnel interface"

event manager applet wan_up
  event track 1 state up
  action 1.0 syslog msg "WAN network has been restored"
  action 2.0 cli command "enable"
  action 3.0 cli command "config t"
  action 4.0 cli command "int tun0"
  action 5.0 cli command "shut"
  action 6.0 cli command "end"
  action 7.0 syslog msg "Returned traffic to the main interface"
7. How to Restrict Certain CLI Commands Without a AAA Server?
The Problem

- Certain users should be limited to a small subset of IOS commands
- While the privilege command can help, we do not want to fiddle with the user privilege levels (plus we’d like some auditing)
- We don’t have a AAA server 😞
The Solution

- Use the enhanced CLI event detector with EEM 4.0 to restrict command usage
  1. Configure EEM to watch all commands

```
event manager applet simpleaaa
event cli pattern ".*" sync yes
```

2. Read list of denied users/commands from a file and write audit results out to a log

```
action 101 file open in flash:denylist r
action 102 file open out flash:cmdhistory a
action 103 file read in cmds
action 104 set tester "$_cli_username"
action 105 append tester " $cli_msg"
action 106 foreach value "$cmds" "\n"
action 107 if $tester eq "$value"
action 108 puts "$_cli_username is not authorized to run $cli_msg"
action 109 file puts out "REJECT: $_cli_username: $cli_msg"
action 110 exit 0
action 111 end
action 112 end
action 113 file puts out "ACCEPT: $_cli_username: $cli_msg"
action 114 exit 1
```
Highlights

- Used the enhanced CLI ED available in EEM 4.0
- Matched on every CLI command executed
  - Applet ran synchronously
  - Returned 0 if the CLI command should be skipped (i.e., not authorized)
  - Returned 1 if the CLI command should be executed (i.e., authorized)
  - Used EEM 4.0 applet-based file manipulation to read the denied commands and write the audit log
8. Send email notification when your home router’s public IP changes
The Problem

- Home router is connected to a public cable network
- The router’s public DHCP address can change periodically
- If the IP changes, we need to know about the new address so we can connect from remote
The Solution

- Use an EEM policy with the *timer* event detector to connect to a remote web site and find the public IP address
- If the address has changed, send the new address in an email
- Cache the new address to prevent sending an email each time the script runs
The Email

from Joe Marcus Clarke
subject New IP Address
to Joe Marcus Clarke
tags Important

New address is 192.168.201.113
The Code

```cisco
::cisco::eem::description "This policy obtains the local (possibly NAT'd) IP address and emails it only if it has changed."
::cisco::eem::event_register_timer tag timer cron cron_entry "0 0 * * *" name ip_addr_timer maxrun 60
::cisco::eem::event_register_none tag none
::cisco::eem::trigger {
    ::cisco::eem::correlate event timer or event none
}

if { ! [info exists ip_addr_file] } {
    puts "ERROR: Policy cannot be run as the ip_addr_file environment variable has not been set"
    exit 1
}

if { ! [info exists _email_server] } {
    puts "ERROR: Policy cannot be run as the _email_server environment variable has not been set"
    exit 1
}

if { ! [info exists _email_from] } {
    puts "ERROR: Policy cannot be run as the _email_from environment variable has not been set"
    exit 1
}

if { ! [info exists _email_to] } {
    puts "ERROR: Policy cannot be run as the _email_to environment variable has not been set"
    exit 1
}

namespace import ::cisco::eem::*
namespace import ::cisco::lib::*
namespace import ::http::*
::http::config -useragent "Mozilla/5.0 (Macintosh; U; Intel Mac OS X 10.6; en; rv:1.9.0.19)"
set tok [:http:geturl "http://www.showmenyip.com/"]
if { [:http:error $tok] != "" } {
    puts "ERROR: Failed to find IP; ":http::error $tok"
    exit 1
}
set addr {}
foreach line [split [:http:data $tok] "\n"] {
    if { [regexp [Your IP address is:
```
if ($addr != "") {
    if (! [catch {set fd [open $ip_addr_file r]} result]) {
        set oldaddr ""
    } else {
        set oldaddr [read $fd]
        close $fd
    }
    if ($addr != $oldaddr) {
        set fd [open $ip_addr_file w]
        puts -nonewline $fd $addr
        close $fd
        set mail_pre "Mailservername: $email_server\n"
        append mail_pre "From: $email_from\n"
        append mail_pre "To: $email_to\n"
        if {[info exists _email_cc]} {
            append mail_pre "Cc: $email_cc\n"
        } else {
            append mail_pre "Cc: \n"
        }
        append mail_pre "Subject: New IP Address\n"
        append mail_pre "New address is $addr\n"
        set mail_msg [uplevel 0 [list subst -nokind -nokind $mail_pre]]
        if [catch {smtp_send_email $mail_msg $email_msg} result] {
            error $result $errorInfo
        }
    }
}
Highlights

- We used the EEM 3.1 description convention to add a bit of documentation to our policy

```
::cisco::eem::description "This policy obtains the local (possibly NAT'd) IP address and emails it only if it has changed."
```

- We used multiple event detectors so the policy can be run periodically and manually for testing

- We used the built-in HTTP client library to fetch our IP address from the internet

```
namespace import ::http::*
```
The New Social Network

Bonus Use Case!

- Traditional CLI interfaces (telnet, SSH) can be boring
- Social networking paradigms such as micro-blogging, social wall, and instant messaging are scalable, secure, and fit into the personal and professional lifestyle of IT users
- What if we embedded social networking features into network devices?
  - Multi-user chat for collaborative troubleshooting and configuration
  - Secure multipoint software upgrades and configuration changes
  - Device-to-device messaging
  - Ubiquitous control and access to network information
  - Natural language interface for interacting with devices
Chatting With Your Network

- XMPP (Jabber) client based on EEM
  - Code in CVS at http://tools.cisco.com/squish/a2a84
- Send CLI to devices
- See network availability in your instant messenger client
Interesting Events on Demand

- XMPP-based publish/subscribe provides events as they happen in your IM client
- Subscribe only to events in which you care about
Is It a Human or Is It a Machine?

- Introduce natural language processing to abstract CLI to a more human-friendly language
- One unifying interface that is intuitive
- The network participates in problem analysis and optimization
The Tweeting Router

- Syslog, SNMP, Email...Tweets
- Use embedded automation to send Twitter updates directly from your devices
- Download the code from [https://supportforums.cisco.com/docs/DOC-19363](https://supportforums.cisco.com/docs/DOC-19363)
EEM Resources
Marketing program driving development of innovative solutions using embedded automation technologies

Virtual team formed by EEM experts from different groups in Cisco

Provides packaged solutions built with EEM and related technologies available for free download

14 EASy solutions available now covering 5 categories, including network management, high availability, security, diagnostics and config automation

http://www.cisco.com/go/easy

ask-easy@cisco.com
EEM Life Cycle Management Using LMS

Policy creation

Improve EEM Policy

EEM Status Reporting

Push Configuration

Monitor EEM via Syslog

Build Configuration

Policy

Configuration

Push

Configuration
Cisco Beyond Has a New Home

Script repository
Over 130+ open source scripts, learn by example

Discussion forums
Ask questions, get answers

Video tutorials

http://www.cisco.com/go/ciscobeyond
EEM Best Practices Doc

• “Living” document at https://supportforums.cisco.com/docs/DOC-12757

• Contains helpful tips and tricks to get the most out of EEM

• Great supplement to the Cisco.com docs
RuBAN suite provides proactive monitoring and network automation using EEM

Provides network auditing services analyzing network health and performance

Offers training on Cisco device instrumentation and management

http://www.davranetworks.com
Tcl Scripting for Cisco IOS

A guide to building and modifying Tcl scripts to automate network administration tasks

Ray Blair, CCIE No. 7050
Arvind Durali, CCIE No. 7016
John Lautmann

ciscopress.com

Recommended Reading

Network-Embedded Management and Applications

Understanding Programmable Networking Infrastructure

Alexander Clemm · Ralf Wolter Editors

Springer
Join Cisco Support Communities!

- **Free** for anyone with Cisco.com registration
- Get **timely** answers to your technical questions
- Find **relevant** technical documentation
- Engage with over 200,000 **top technical experts**
- **Seamless** transition from discussion to TAC Service Request (*Cisco customers and partners only*)

The Cisco Support Community is your one-stop community destination from Cisco for sharing current, real-world technical support knowledge with peers and experts.

supportforums.cisco.com
supportforums.cisco.mobi
Call to Action…

Visit the World of Solutions:-

- Cisco Campus
- Walk-in Labs
- Technical Solutions Clinics

- Meet the Engineer

- Lunch Time Table Topics, held in the main Catering Hall

- Recommended Reading: For reading material and further resources for this session, please visit [www.pearson-books.com/CLMilan2014](http://www.pearson-books.com/CLMilan2014)
Complete Your Online Session Evaluation

- Complete your online session evaluation
- Complete four session evaluations and the overall conference evaluation to receive your Cisco Live T-shirt