SEND WEBEX NOTIFICATION WHEN CHANGE IS MADE

This is EEM applet with Python script which triggers webex notification, everytime change is made on device.

The eem script leverage Embedded Event Manager (EEM) technology to monitor syslog messages generated by the device for the specified syslog ID: SYS-5-CONFIG_I. This syslog message is generated every time a configuration change is made on the device.

When the EEM script detects a syslog message:

```
%SYS-5-CONFIG_I: Configured from console by cisco on vty1 (10.61.95.152)
```

it would initiate a notification to be sent via Webex. The notification would contain information about the device that triggered the syslog message and information about who did the change (user) and from where this change was made (source ip address of user)

The EEM applet would be programmed to continuously monitor the device for the specified syslog message and to trigger the Webex notification process each time it detects a new configuration change event.

Webex Notifications are being sent by Python script inside GuestShell.

EEM applet

```
event manager applet CONFIG_CHANGED
event syslog pattern "%SYS-5-CONFIG_I: Configured from console by.*"
action 1.0 regexp "\n*from console by ([^\s]+)" $_syslog_msg match user
action 2.0 regexp "\n*from console by.*\(([^\)]+)" $_syslog_msg match from
action 3.0 set text $user,$from
action 4.0 cli command "en"
action 5.0 cli command "guestshell run python3 message.py <ACCESS_TOKEN>
kmazurki@cisco.com --text $text"
exit
exit
```

`<ACCESS_TOKEN>` - it can either be personal user access token, or webex bot can be created on https://developer.webex.com

Python script

```
import argparse
import requests
import json
import cli
import os
import re

def send_message(text,token,toPersonEmail,hostname):
    url = "https://webexapis.com/v1/messages"

    payload = json.dumps({
        "toPersonEmail": toPersonEmail,
        "markdown": f"Host: **{hostname}** -> **%SYS-5-CONFIG_I:** Configured from console by **{text.split(',')[0]}** from **{text.split(',')[1]}**"
    })

    headers = {
        'Authorization': f'Bearer {token}',
        'Content-Type': 'application/json'
    }


    response = requests.request("POST", url, headers=headers, data=payload)

    return response.json()

def main():

    # Get hostname of device
    hostname = cli.cli('show run | in hostname').split('\n')[0].split(' ')[1]

    # Create the parser object
    parser = argparse.ArgumentParser(description='A script which sends message to webex')

    # Add a required argument for token
    parser.add_argument('token', help='Personal Access Token')

    # Add a required argument for toPersonMail
    parser.add_argument('toPersonEmail', help='Email of person you want to send webex message to')

    # Add a required argument for text
    parser.add_argument('--text', help='Message text')

    # Parse the arguments
    args = parser.parse_args()

    # Send message
    message = send_message(token=args.token,
toPersonEmail=args.toPersonEmail, text=args.text,hostname=hostname)

    # Print the greeting
    print(f'{message}')

if __name__ == '__main__':
    main()
```

Note. `debug EEM`

```
terminal monitor
debug event manager action cli
```