opEvents - Syslog Handling - Adding a New Format

Purpose

This page will explain how to add a new syslog format in the event the default settings are not handling the syslog messages properly. Consider a situation where a new type of device has been added to the network.

For this discussion well use the term 'newVendor' to be the variable that represents the new format we want opEvents to handle.

Steps

- Choose a unique syslog facility for the newVendor.
- Provision rsyslog to handle the syslog messages appropriately.
- Provision opEvents to parse and process the syslog messages.

rsyslog Provisioning- nmis8

Determine what facility level these syslog messages should be stamped with. The syslog server will key on this facility level in order to send the syslog message to the proper file. If the device syslog is very similar to Cisco then you may want to simply use the local7 facility and the syslog messages will be sent to /usr/local/nmis8/logs/cisco.log. Configure the nodes in question to send syslog to NMIS at the proper facility level. For this example we will use local6 for newVendor. Typically facilities local0 through local7 are used for processing syslog from external nodes.

Ensure the syslog server is provisioned to received messages (udp & tcp). This configuration is below and can be made on the /etc/rsyslog.conf file.

```
### /etc/rsyslog.conf

# enable network sources
module(load="imudp")
input(type="imudp" port="514")

module(load="imtcp" MaxSessions="1000" MaxListeners="50")
input(type="imtcp" port="514"

# and handle inbound/poller NMIS syslogs
local7.*                /usr/local/nmis8/logs/cisco.log
local1.*                /usr/local/nmis8/logs/poller_event.log
```

Next we'll tell rsyslog where to file messages that arrive with the facility local6. (rsyslog can route messages based on many different attributes and has several other configuration options such as inserting a local timestamp. There is also the ability to place custom rsyslog configuration in any file with a .conf extension within the /etc/rsyslog.d/ directory. More information regarding rsyslog may be found here: http://www.rsyslog.com/doc/master/index.html)

```
### /etc/rsyslog.conf

# and handle inbound/poller NMIS syslogs
local7.*                /usr/local/nmis8/logs/cisco.log
local6.*                /usr/local/nmis8/logs/newVendor.log
local1.*                /usr/local/nmis8/logs/poller_event.log
```

After modifying /etc/rsyslog.conf the syslog daemon must be restarted.

```
[root@opmantek ~]# service rsyslog restart
Shutting down system logger:                               [  OK  ]
Starting system logger:                                    [  OK  ]
```

Now when syslog messages are received with facility level local6 we will see them in the /user/local/nmis8/logs/newVendor.log file. If this file does not exist it will be created automatically.

rsyslog Provisioning- nmis9

Determine what facility level these syslog messages should be stamped with. The syslog server will key on this facility level in order to send the syslog message to the proper file. If the device syslog is very similar to Cisco then you may want to simply use the local7 facility and the syslog messages will be sent to /usr/local/nmis9/logs/cisco.log by default. Configure the nodes in question to send syslog to NMIS at the proper facility level. For this example we will use local6 for newVendor. Typically facilities local0 through local7 are used for processing syslog from external nodes.

Ensure the syslog server is provisioned to received messages (udp & tcp). This configuration is below and can be made on the /etc/rsyslog.conf file.

```
### /etc/rsyslog.conf

# enable network sources
module(load="imudp")
input(type="imudp" port="514")

module(load="imtcp" MaxSessions="1000" MaxListeners="50")
input(type="imtcp" port="514"

# and handle inbound/poller NMIS syslogs
local7.*                /usr/local/nmis9/logs/cisco.log
local1.*                /usr/local/nmis9/logs/poller_event.log
```

Next we'll tell rsyslog where to file messages that arrive with the facility local6. (rsyslog can route messages based on many different attributes and has several other configuration options such as inserting a local timestamp. There is also the ability to place custom rsyslog configuration in any file with a .conf extension within the /etc/rsyslog.d/ directory. More information regarding rsyslog may be found here: http://www.rsyslog.com/doc/master/index.html)

```
### /etc/rsyslog.conf

# and handle inbound/poller NMIS syslogs
local7.*                /usr/local/nmis9/logs/cisco.log
local6.*                /usr/local/nmis9/logs/newVendor.log
local1.*                /usr/local/nmis9/logs/poller_event.log
```

After modifying /etc/rsyslog.conf the syslog daemon must be restarted.

```
[root@opmantek ~]# service rsyslog restart
Shutting down system logger:                               [  OK  ]
Starting system logger:                                    [  OK  ]
```

Now when syslog messages are received with facility level local6 we will see them in the /user/local/nmis9/logs/newVendor.log file. If this file does not exist it will be created automatically.

opEvents3 Provisioning

For the sake of this discussion let's assume the new vendor can be parsed with the existing cisco_alternate rules found in /usr/local/omk/conf /EventParserRules.nmis. We need to tell opEvents to use these parser rules on /usr/local/nmis8/logs/newVendor.log. This is done by modifying /usr/local /omk/conf/opCommon.nmis. Find the 'opevents_logs section and add the 'cisco_alternate', '<nmis_logs>/newVendor' relationship.

```
### /usr/local/omk/conf/opCommon.nmis

    'opevents_logs' => {
      'cisco_alternate' => [
        '<nmis_logs>/newVendor.log'
      ],
      'cisco_syslog' => [
        '<nmis_logs>/cisco.log'
      ],
      'nmis_eventlog' => [
        '<nmis_logs>/event.log'
      ],
```

After modifying opCommon.json the opEvents daemon must be restarted.

```
[root@opmantek ~]# service opeventsd restart
Restarting opevents daemon opeventsd                       [  OK  ]
[root@opmantek ~]#
```

Create an event action policy as described here: Event Actions and Escalation (opEvents 4)

Once these actions are complete the syslog messages from newVendor should be seen in opEvents.

opEvents4 Provisioning

For the sake of this discussion let's assume the new vendor can be parsed with the existing cisco_alternate rules found in /usr/local/omk/conf /EventParserRules.json. We need to tell opEvents to use these parser rules on /usr/local/nmis9/logs/newVendor.log. This is done by modifying /usr/local /omk/conf/opCommon.json. Find the 'opevents_logs section and add the 'cisco_alternate', '<nmis_logs>/newVendor' relationship.

```
### /usr/local/omk/conf/opCommon.json

    'opevents_logs' => {
      'cisco_alternate' => [
        '<nmis_logs>/newVendor.log'
      ],
      'cisco_syslog' => [
        '<nmis_logs>/cisco.log'
      ],
      'nmis_eventlog' => [
        '<nmis_logs>/event.log'
      ],
```

After modifying opCommon.json the opEvents daemon must be restarted.

```
[root@opmantek ~]# service opeventsd restart
Restarting opevents daemon opeventsd                       [  OK  ]
[root@opmantek ~]# service omkd restart
```

Create an event action policy as described here: Event Actions and Escalation (opEvents 4) or Event Actions and Escalation (opEvents 3)

Once these actions are complete the syslog messages from newVendor should be seen in opEvents.

Related Topics

- opEvents - Syslog Handling - Adding a New Vendor
- opEvents - Centralized Logging Solution
- SNMP Traps with Cisco and Other devices
- High Volume SNMP Trap Processing