Opmantek Synergy - opEvents Triggers opConfig to Collect Targeted Diagnostic Data

Purpose

This article will provide an example of opConfig collecting specific command output based on event characteristics. The command output becomes embedded into the event allowing network operators to more quickly identify root cause and affect resolution.

Use Case

For this example we'll take actions if an event's stateful element is a BGP Peer. opEvents will fire scripts that result in useful information for engineers to follow up on.

Related Pages

- Event Actions and Escalation
- opConfig Configuration Guide (Version 1.x)

Configuration

opEvents

In this case the necessary configuration for opEvents is contained in /usr/local/omk/conf/EventActions.nmis. There are two sections we will be concerned with; script and policy.

Let's have a look at the script section first. For this example we want to ping the node that reported the problem, the BGP peer it's having an issue with, and leverage opConfig to gather related command output. Notice the arguments passed to opconfig-cli.pl, there is a command set name; IOS_TS_BGP. In the opConfig step we'll define what these specific router commands are.

/usr/local/omk/conf/EventActions.nmis

```
        'script' => {
              'ping_node' => {
                      arguments => '-c 5 node.host',
                      exec => '/bin/ping',
                      output => 'save'
               },
               'ping_neighbor' => {
                      arguments => '-c 5 event.element',
                      exec => '/bin/ping',
                      output => 'save'
               },
               'troubleshoot_bgp' => {
                      arguments => 'act=run_command_sets names=IOS_TS_BGP nodes=node.name
print_command_output=true mthread=false debug=0 quiet=1',
                      exec => '/usr/local/omk/bin/opconfig-cli.pl',
                      output => 'save'
               },
       }
```

In the policy section we define a rule that matches stateful properties that contain 'BGP Peer'. If there's a match it will fire the scripts defined above.

/usr/local/omk/conf/EventActions.nmis

```
        'policy' => {
               '10' => {
                       IF => 'event.any',
                       THEN => {
                               '10' => {
                                       IF => 'event.stateful =~ "BGP Peer"',
                                       THEN => 'priority(8) AND script.ping_node() AND script.ping_neighbor()
AND script.troubleshoot_bgp()',
                                       BREAK => 'false'
                               },
                       },
         }
```

opConfig

The /usr/local/omk/conf/command_sets.d directory is where we have files detailing exec commands to be executed on nodes. For this example we are concerned with ios.nmis. Our IOS_TS_BGP command set will be running the following commands.

- show ip bgp
- show ip bgp summary
- show ip route summary

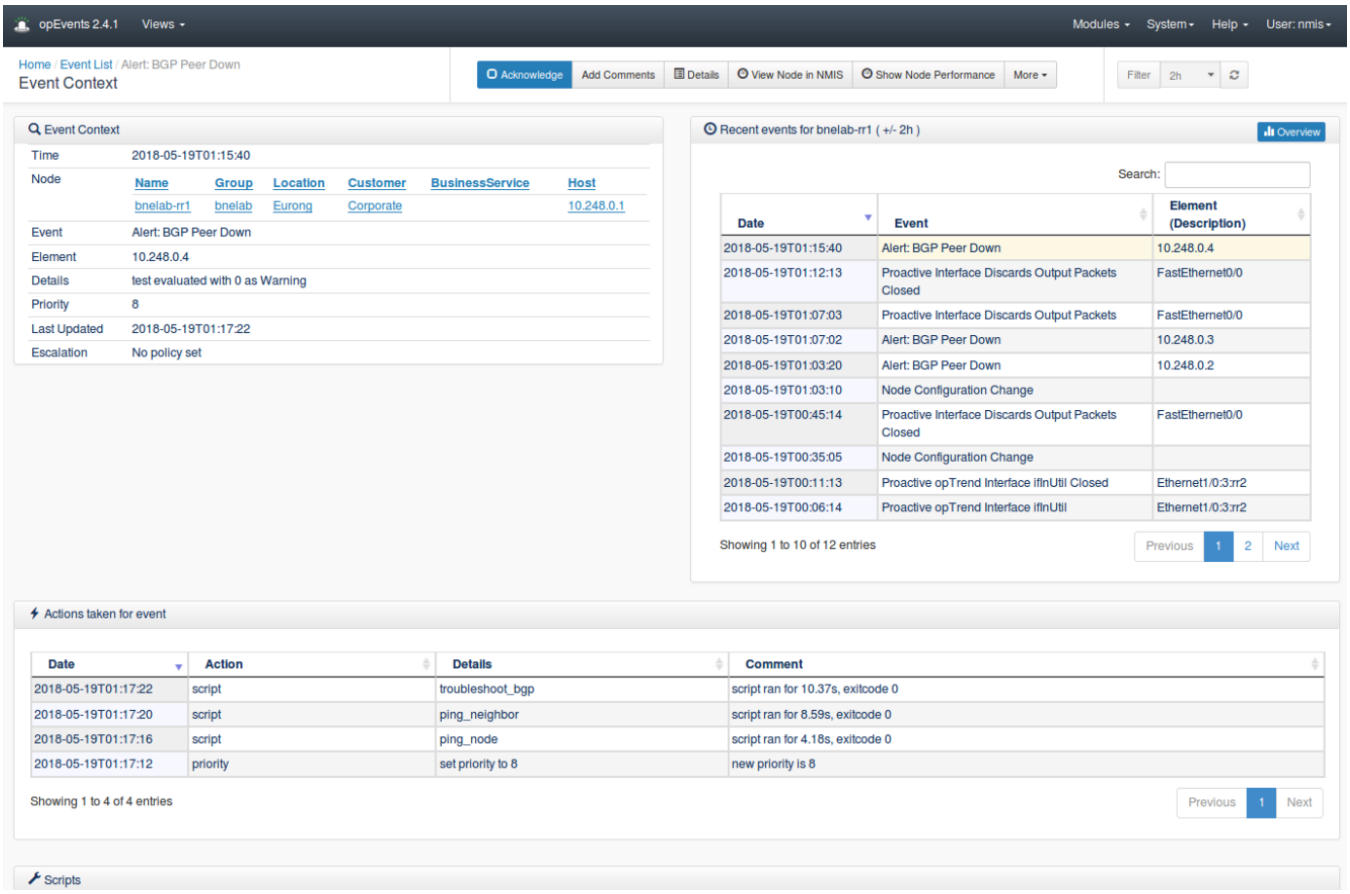
If you have a large number of BGP routes we do not recommended 'show ip bgp'.

/usr/local/omk/conf/command_sets.d/ios.nmis

```
  'IOS_TS_BGP' => {

    'os_info' => {
    'version' => '/12.2|12.4|15.\d+/',
    'os' => 'IOS',
   },
    'scheduling_info' => {
    'run_commands_on_separate_connection' => 'false',
    'attempt_timeout_recovery' => 1,
   },

   'purging_policy' => {
    'keep_last' => 1000,
    'purge_older_than' => 2592000, # 30 days
    'autoprotect_first_revision' => 'true',
   },
   'commands' => [
                      {
      'multipage' => 'true',
      'privileged' => 'true',
      'command' => 'show ip bgp',
      'tags' => [ 'troubleshooting', 'routing', 'detect-change' ],
    },

    {
      'multipage' => 'true',
      'privileged' => 'true',
      'command' => 'show ip bgp summary',
      'tags' => [ 'troubleshooting', 'routing', 'detect-change' ],
    },

    {
      'multipage' => 'true',
      'privileged' => 'true',
      'command' => 'show ip route summary',
      'tags' => [ 'troubleshooting', 'routing', 'detect-change' ],
    },
   ],
 },
```

Testing and Verification

The easiest way to test this configuration is to administratively shutdown a BGP peer. After the next NMIS collect cycle a BGP Peer Down alert will be processed by opEvents. Here's an example from our lab.



Notice the Actions section is notifying us that our scripts fired. Scrolling down we can view the script output; it's now embedded into the event.

```
Command Output for show ip bgp summary:
BGP router identifier 10.248.5.41, local AS number 65001
BGP table version is 43, main routing table version 43
42 network entries using 5376 bytes of memory
84 path entries using 4368 bytes of memory
8/8 BGP path/bestpath attribute entries using 992 bytes of memory
4 BGP AS-PATH entries using 96 bytes of memory
0 BGP route-map cache entries using 0 bytes of memory
0 BGP filter-list cache entries using 0 bytes of memory
BGP using 10832 total bytes of memory
BGP activity 42/0 prefixes, 126/42 paths, scan interval 60 secs

Neighbor        V           AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
10.248.0.2      4        65001       0       0        1    0    0 00:14:49 Idle (Admin)
10.248.0.3      4        65001       0       0        1    0    0 00:10:48 Idle (Admin)
10.248.0.4      4        65001       0       0        1    0    0 00:02:37 Idle (Admin)
10.248.0.5      4        65001    2823    2840       43    0    0 1d18h           0
10.248.0.6      4        65001    2827    2837       43    0    0 1d18h           0
10.248.0.7      4        65001    2835    2828       43    0    0 1d18h          21
10.248.0.8      4        65001    2828    2834       43    0    0 1d18h          21
10.248.0.9      4        65001    2832    2829       43    0    0 1d18h          21
10.248.0.10     4        65001    2834    2831       43    0    0 1d18h          21

Command Output for show ip route summary:
IP routing table name is default (0x0)
IP routing table maximum-paths is 32
Route Source    Networks    Subnets     Replicates  Overhead    Memory (bytes)
connected       0           30          0           1840        5160
static          1           1           0           120         344
ospf 1          0           108         0           10200       19008
  Intra-area: 108 Inter-area: 0 External-1: 0 External-2: 0
  NSSA External-1: 0 NSSA External-2: 0
bgp 65001       0           33          0           1980        5676
  External: 0 Internal: 33 Local: 0
internal        1                                               7064
Total           2           172         0           14140       37252
```

ping_neighbor (completed at 2018-05-19T01:17:20, exit code 0)

```
PING 10.248.0.4 (10.248.0.4) 56(84) bytes of data.
64 bytes from 10.248.0.4: icmp_seq=1 ttl=251 time=2284 ms
64 bytes from 10.248.0.4: icmp_seq=2 ttl=251 time=1881 ms
64 bytes from 10.248.0.4: icmp_seq=3 ttl=251 time=1992 ms
64 bytes from 10.248.0.4: icmp_seq=4 ttl=251 time=2063 ms

--- 10.248.0.4 ping statistics ---
5 packets transmitted, 4 received, 20% packet loss, time 4009ms
rtt min/avg/max/mdev = 1881.760/2055.583/2284.129/146.998 ms, pipe 3
```