BOF: Nagios Plugins New Threshold Specification Syntax

Comments to :
nagiosplug-devel@lists.sourceforge.net

October 1, 2013

Nagios®
World Conference
North America

Saint Paul, MN

Current State of Nagios Plugins Syntax

There is Nagios Threshold Specification as part of Nagios Plugins Development Guidelines:

https://www.nagios-plugins.org/doc/guidelines.html

This is supported by most core plugins and followed by many other plugin authors. However:

- The range syntax is confusing and quite limited. People think the @a:b syntax for checking if something is within the range is unnatural and difficult to understand.
- Many (most!) plugins have multiple metrics that can be checked, yet nagios development guidelines call for threshold to be specified in one set of –warn (-w) and –critical (-c) options.
- The syntax is not extendable for adding additional parameters related to checking of specific metric. Many plugins end up with a ton of options which becomes even more confusing for the users

History of the proposal

- To encourage discussion on how to deal with these issues a draft proposal called “New Threshold Syntax” was written by Ton Voon five years ago. It used to be available at nagiosplugins.org
- The Plugins Development Team discussed this internally, but didn't reach a consensus on how to proceed, so this was never implemented by any core plugin. Ton is no longer involved in Nagios-Plugins project either.
- However some 3rd-party plugin authors liked this proposal and moved forward to partially implement it in their own libraries and plugins. Current libraries that partially support this or something similar are: Pynag (http://pynag.org/),
  JNRPE (http://jnrpe.sourceforge.net/),
  Naglio (https://github.com/willixix/WL-NagiosPlugins)
- Current proposal is an extension of original with some additions from Naglio.pm plus ideas proposed at nagios-plugins mail list discussion. Current text is at:
  https://github.com/willixix/nagios-plugins/wiki/New-Threshold-Syntax

Proposal: New Threshold Option

https://github.com/willixix/nagios-plugins/wiki/New-Threshold-Syntax

- New single Threshold option that can be repeated:

```
--threshold={threshold definition} --th={threshold definition}
```

- The threshold definition is a subgetopt format with a list of keywords followed by '=' or ':' and then data.

```
--threshold metric={metric},name={name},label={label},ok={range},
warn={range},crit={range},absent=critical|warning|ok|unknown,
display=yes|no,perf=yes|no,unit={unit},prefix={SI prefix}
```

OR

```
--threshold=metric:{metric},name:{name},label:{label},ok:{range},warn:
{range},crit:{range},absent:critical|warning|ok|unknown,display:yes|no,
perf:yes|no,unit:{unit},prefix:{SI prefix}
```

Examples

- To check httpd processes are OK if the virtual size is under 8096 bytes. Warn until they reach 16182, but bigger than that is CRITICAL.

  old: check_procs -w 8096 -c 16182 -C httpd --metric VSZ

  new: check_procs -C httpd –th metric=vsize,ok=0..8096,warn=8097..16182

- There should always be one and only one 'tnslsnr' process. Otherwise critical.

  old: check_procs -w 1:1 -c 1:1 -C tnslsnr

  new: check_procs -C tnslsnr --th metric=count,ok=1..1

- Load averages (1,5,15 minute) should be within reasonable ranges.

  old: check_load -w 1.0,0.8,0.7 -c 1.5,1.3,1.0

  new: check_load --th metric=1min,ok=0..1.0,warn=1.0..1.5 --th metric=5min,ok=0..0.8,warn=0.8..1.3 --th metric=15min,ok=0..0.7,warn=0.7..1.0

Proposal: New Range Syntax

- New “natural” format for range. Simple ranges have format:

  start..end (such as “warn=0..20”)

  - start and end are numeric or “inf”. require start ≤ end
  - if start = "inf" or “-inf”, this is negative infinity.
  - if end = "inf", this is positive infinity
  - alert is raised if value is inside start and end range

- Complex ranges have the following additions:

  addition of () and [] brackets which are used in math for interval notation
  () is open interval, means end points are not included
  [] is closed interval, means end points are included
  addition of ^ used to negate interval, preferably this is followed by () or []

- Examples (“current range format” → “new range format”):

10	→	10 or ^[0..10]	- alert if < 0 or > 10, (outside of {0 .. 10})
10:	→	^[10..inf]	- alert if < 10, (outside of {10 .. ∞})
~:10	→	^[-inf..10]	- alert if > 10, (outside of {-∞ .. 10})
10:20	→	^[10..20]	- alert if < 10 or > 20, (outside of {10 .. 20})
@10:20	→	10..20	- alert if ≥ 10 and ≤ 20, (inside of {10 .. 20})

Proposal: 'warn', 'crit', 'ok' levels

- “ok”, “warn”, “crit” are level specification keywords. Basic syntax:

  ok:{range},warn:{range},crit:{range}   OR
  ok={range},warn={range},crit={range}

- Ranges are used with warn, crit, ok keywords which combined specify when and what type of alert is to be issued:

  - if no levels are specified, return OK
  - If an ok level is specified and value is within range, return OK
  - If a critical level is specified and value is within range, return CRITICAL
  - If a warning level is specified and value is within range, return WARNING
  - If an ok level is specified, return CRITICAL
  - Otherwise return OK

- These keywords maybe repeated more than once to specify complex non-continuous range. The ranges are combined as a logical “OR”.

- A proposed extension is to also have “awarn”, “aok”, “acrit” keywords which allow levels to be combined with a logical AND. The primary use of this is with multiple metrics specifying that alert is to be issued when values for both metrics are within specified ranges.

Proposal: 'metric', 'name', 'regex', 'label' keywords

- metric={name} OR metric:{name}

  This keyword specifies alphanumeric name of the metric being checked by the plugin (name has meaning for plugin itself).This is usually first keyword in threshold option line

- name={name} OR name:{name}

  This is used with plugins that can get data on multiple named data values (attributes) and where different metric to be checked are available for these named attributes. Examples of this are:
  - process name, where process memory and cpu use are metrics.
  - network interface name, where bytes_in and bytes_out are metrics.

- regex='yes|no' OR regex:'yes|no'

  Enables or disables regex matching of {name} in 'name' and 'metric'

- label={name}' OR label:{name}
  perf_label={name} OR perf_label:{name}

  These are used re-label 'name' or 'metric' for status line and perf output, usually to make long name shorter.

Proposal: other keywords

- absent='warn|critical|ok|unknown'
  Nagios exit status and alert if specified named value or metric are not available. This overrides default 'unknown' many plugins use.

- display='yes|no' OR display:'yes|no'
  Specifies if metric info & value should be included in status line output

- perf='yes|no' OR perf:'yes|no'
  Specifies if metric value should be included in perf output

- prefix={prefix} OR prefix:{prefix}
  The prefix is used to multiply the input range for display of data in status output. Allowed {prefix} values are slight expansion of those defined by NIST at http://physics.nist.gov/cuu/Units/prefixes.html:

- unit={unit} OR uom={unit} OR unit:{unit} OR uom:{unit}
  {unit} is a single symbol or string that specify the Unit of Measurement (UOM) for plugins that do not know about the type of value returned. Base UOMs are: '%' for percent and 'c' for performance counters.. Custom UOMs may include one-letter prefix followed by abbreviation or full NIST prefix followed by '-' and then label. For example:
  kb = kilo-b = kilo-bits

Discussion

- Is this good proposal base or are there serious issues?
- The proposal currently does mention how to output new complex ranges for use by performance graphing utilities. Any suggestions on how to best handle this?
- It may take some time for official plugin developers to implement especially since Ton Voon is no longer involved. Should we ask them to do it?
- Do we want to move forward and and finalize this as standard?
- Any other suggestions for plugin standardization?

Extra: XML spec for describing plugins?

- Separate issue how to provide information on how to configure various plugins for web user interface that various nagios packages have.
- One proposal could be to write an XML spec that could be used to describe options that plugins have. This can be thought of as structured help (and in fact such XML can be converted into manual page or help). This would be something like:

```
<plugin name=”check_snmp_..”>
  <options>
     <option name=”SNMP_Auth_Protocols”>
       <option_lines><line>L</line><line>authprotocols</line></option_lines>
       <accepted_values_list>
           <value id=1>md5</value><value id=2>sha1</value>
       </accepted_values_list>
     </option>
 </options>
</plugin>
```

- For any Nagios package vendors are here - would this be of any interest to you?