NAME

Smokeping::probes::TCPPing – TCPPing Probe for SmokePing

SYNOPSIS

```
*** Probes ***


+TCPPing


binary = /usr/bin/tcpping # mandatory
forks = 5
offset = 50%
step = 300
timeout = 15


# The following variables can be overridden in each target section
pings = 5
port = 80


# [...]


*** Targets ***


probe = TCPPing # if this should be the default probe


# [...]


+ mytarget
# probe = TCPPing # if the default probe is something else
host = my.host
pings = 5
port = 80
```

DESCRIPTION

Integrates TCPPing as a probe into smokeping. The variable **binary** must point to your copy of the TCPPing program. If it is not installed on your system yet, you can get it from http://www.vdberg.org/˜richard/tcpping. You can also get it from http://www.darkskies.za.net/˜norman/scripts/tcpping.

The (optional) port option lets you configure the port for the pings sent. The TCPPing manpage has the following to say on this topic:

The problem is that with the widespread use of firewalls on the modern Internet, many of the packets that **traceroute** (8) sends out end up being filtered, making it impossible to completely trace the path to the destination. However, in many cases, these firewalls will permit inbound TCP packets to specific ports that hosts sitting behind the firewall are listening for connections on. By sending out TCP SYN packets instead of UDP or ICMP ECHO packets, tcptraceroute is able to bypass the most common firewall filters.

It is worth noting that tcptraceroute never completely establishes a TCP connection with the destination host. If the host is not listening for incoming connections, it will respond with an RST indicating that the port is closed. If the host instead responds with a SYN|ACK, the port is known to be open, and an RST is sent by the kernel tcptraceroute is running on to tear down the connection without completing three-way handshake. This is the same half-open scanning technique that **nmap** (1) uses when passed the −sS flag.

VARIABLES

Supported probe-specific variables:

binary

The location of your tcpping script.

Example value: /usr/bin/tcpping

This setting is mandatory.

forks
>   Run this many concurrent processes at maximum

>   Example value: 5

>   Default value: 5

offset
>   If you run many probes concurrently you may want to prevent them from hitting your network all at the same time. Using the probe-specific offset parameter you can change the point in time when each probe will be run. Offset is specified in % of total interval, or alternatively as 'random', and the offset from the 'General' section is used if nothing is specified here. Note that this does NOT influence the rrds itself, it is just a matter of when data acqusition is initiated. (This variable is only applicable if the variable 'concurrentprobes' is set in the 'General' section.)

>   Example value: 50%

step
>   Duration of the base interval that this probe should use, if different from the one specified in the 'Database' section. Note that the step in the RRD files is fixed when they are originally generated, and if you change the step parameter afterwards, you'll have to delete the old RRD files or somehow convert them. (This variable is only applicable if the variable 'concurrentprobes' is set in the 'General' section.)

>   Example value: 300

timeout
>   How long a single 'ping' takes at maximum

>   Example value: 15

>   Default value: 5

Supported target-specific variables:

pings
>   How many pings should be sent to each target, if different from the global value specified in the Database section. Note that the number of pings in the RRD files is fixed when they are originally generated, and if you change this parameter afterwards, you'll have to delete the old RRD files or somehow convert them.

>   Example value: 5

port
>   The TCP port the probe should measure.

>   Example value: 80

AUTHORS

>   Norman Rasmussen <norman AT rasmussen DOT co DOT za> Patched for Smokeping 2.x compatibility by Anton Chernev <maznio AT doom DOT bg>