Installation Instructions for the fping feeder for Nagios®

in a GroundWork Monitor Server

Version 3.0

December 9, 2007

Introduction

This paper describes the setup of the fping feeder. It is a script that runs the fping tool (fping.sourceforge.net) to ping all the hosts in a GroundWork Monitor installation, and feeds the results to the Nagios® command pipe. It can be run either as a daemon feeder or as a plugin.

Ping testing hosts externally to Nagios® is a good idea, because the active host checks in Nagios 1.x and 2.x are "blocking" checks – Nagios pauses all other activity while doing them, and this can cause latency issues when a lot of hosts go down at once. This feeder checks **every** host with a ping, in groups of 100 or so at a time, in a continuous cycle starting typically every 5 minutes, and can therefore do this function for Nagios. The feeder function can also be scheduled as a plugin, and will then run periodically as determined by the Nagios scheduler.

The hosts to check are pulled from the GroundWork Monitor Architect (monarch) database automatically. Results are then submitted to Nagios as passive checks, via the command pipe. This feeder will also submit service checks for specific service names, which will be processed only if they exist. The services it looks for are "ping only" services, based on the service name, for example "icmp_ping". See the script code itself for details.

Performance data is taken from the fping run, using the fping -e option, which will return the round trip time in milliseconds. Performance data is only used for service checks, and is not submitted for host checks.

GroundWork Monitor (Open Source and Professional) includes the supervise utility, which keeps programs running (restarts them when they crash). This is how all the GroundWork "feeder" programs are installed. This package installs the fping feeder under supervise, with an initial configuration set to disable its operation. You can also call this script as a Nagios plugin with the --plugin or -p option, which will execute a single ping sweep of all hosts, and then exit.

If you want to use another method, feel free to customize this code. GroundWork support service people know the supervise and plugin methods, however, and will insist that you use one of these if you want to get support for it.

Need to verify: Version 3.0 supports the ability to "walk" the Nagios dependency tree, and submit host status that matches the Nagios host-dependency calculation. The logic is identical to that in Nagios, in that if a host is down, and it has no parents, it will get a down status assigned by this feeder. If it has at least one parent that is up, it will likewise be set to down, but if all the parents are down or unreachable, it will be set to unreachable, thus emulating the dependency logic that Nagios uses. Setting the host notification options to down and recovery, and NOT setting notification for unreachable status will effectively suppress alarm storms.

Version 3.0 also supports several means to smooth out the processing. There is a default 0.1-second interval between the spawnings of successive batches of host pings, so as not to overload the system with a forking storm. There is also a limit on the number of host check results that can be submitted at once, using a default batch of 500, using as many additional batches as required. These parameters may be changed by editing the configuration file. An earlier modification of this type was introduced when a customer encountered an issue with 1700+ host results submitted at once, causing a segfault in Nagios. This has likely been addressed in later versions of Nagios, but the capability still remains as a precaution.

Release Notes

Version 3.0 provides the following enhancements:

- This release is packaged as an RPM (groundwork-fping-feeder), for deployment on a GroundWork Monitor 5.1.X system. A tarball version is also available.
- As part of the re-packaging, the installation process has changed. The software is installed in its final target location as a feeder, but it is disabled in the initial configuration file so it will not run until it has been locally configured.
- The fping processing script now emits a final service check result for the fping_process service, so the state of this processing can be tracked and alarmed on if it runs too long. This service is associated with the host on which the script runs.
- Support has been added for running the fping processing on a child server, only probing the corresponding subset of the hosts being monitored by the overall setup.
- As part of the child-server support, the fping check results can be optionally directed to two separate target locations, fed via NSCA, so both parent and child servers can be updated from the same probing activity.
- The pause time between successive fping invocations has been generalized to allow fractional-second sleeps, to minimize the latency of reported results.
- The size of batches of hosts on fping commands has been decoupled from the size of batches of results sent to the monitoring server via NSCA, to provide greater operating efficiency.
- Ping round-trip times are now properly reported as both host-check and service-check result performance metrics ("rta", supposedly the round-trip average time (ms) for pings, but in practice it may represent the round-trip time for only the last, successful ping to each host).
- Support has been added back for direct writing to a local Nagios command pipe, for configurations where that makes sense.
- The daemon feeder run under gwservices is no longer set to execute as root.

Prerequisites

1. GroundWork Monitor Open Source or Professional, v 4.5 or above.
   Note: You could use this feeder for Nagios alone, if you are also using the GroundWork Monitor Architect ("monarch" database). You would need to adjust the installation to fit how you installed Nagios, and you would need to work out how to daemonize the script yourself. The GroundWork Monitor Open Source package includes everything you need.
2. fping (fping.sourceforge.net) installed in /usr/local/groundwork/bin . The binary is supplied as /usr/local/groundwork/sbin/fping in the GroundWork Monitor 5.1.X release; it can be copied from there.
3. This package of the fping_process.pl script, its fping_process.conf configuration file, and the feeder-nagios-fping supervise service. It is available in both RPM and tarball forms. Note that you could create the supervise service yourself with the appropriate commands, if you were not installing via RPM or the install script contained in the tarball.
4. Installation of this package requires prior installation of the groundwork-perl-config-general and groundwork-perl-typedconfig packages, in that order.

Installation

Once you have installed the other packages, download the appropriate feeder package RPM from GroundWork Connect, with a filename like:

groundwork-fping-feeder-3.0.0-2582.noarch.rpm

Copy the RPM file to the GroundWork server (using scp, or any method that works).

Install the RPM while logged in as root, with:

rpm -Uvh groundwork-fping-feeder-3.0.0-*buildnumber*.noarch.rpm

Decide whether you will run the feeder as a plugin or as a daemon. The standard installation process installs the daemon service, ready to run as soon as the configuration file is properly localized and processing is enabled. If you want to run the feeder as a plugin, you will need to disable running the daemon, by renaming /usr/local/groundwork/services/feeder-nagios-fping/run as run.original so it is not found by gwservices and the daemon will not be started in that context.

Then modify the configuration options in the /usr/local/groundwork/etc/fping_process.conf file, following the comments in that file and in the *Tuning and Options* section below. The last value to change is the enable_processing setting, so an invocation of the feeder will be effective.

If you run the feeder as a plugin, set it up in GroundWork Monitor to run periodically that way. If you run it as a daemon, you may need to kill the existing copy of the fping_process.pl process, to force a new copy to be spawned by gwservices so it will read the modified configuration file.

Manual Installation

The rest of the notes in this section may be incomplete or slightly wrong.

For a non-RPM install, where the software is supplied as a tarball rather than as an RPM, the installation can be completed by just running (as root) the included “install_fping.sh” script and answering the prompts. The remainder of this section simply describes what that script does, for the benefit of those using platforms where RPMs are not yet provided, or for older releases of GroundWork Monitor where the fping binary is not provided.

There are two parts to the installation: the fping binary, and the feeder package.

Check if the file /usr/local/groundwork/bin/fping exists before proceeding with installation. If it is not present, your options are:

- GroundWork Monitor 5.1 and above includes the fping binary, in the sbin directory (i.e., as /usr/local/groundwork/sbin/fping). It must be copied to the parallel bin directory (i.e., to /usr/local/groundwork/bin/fping) for actual use.
- The binaries for RHEL4 32-bit and RHEL4 64-bit platforms are available with the download for this package (as fping.rhel4.i386 and fping.rhel4_64.x86_64, respectively).
- Build from source. Download and install fping according to the instructions on its web page (http://fping.sourceforge.net/).

Regardless of the source of your copy, install the fping binary in /usr/local/groundwork/bin, set the owner to root, group to nagios, and make the file setuid root. It is very important that the fping binary is setuid root, or the feeder will not work.

Check that the binary is owned by root.nagios, and setuid root:

ls -l /usr/local/groundwork/bin/fping

You should see something like:

-rwsr-sr-x 1 root nagios 61348 Jan 2 14:54 /usr/local/groundwork/bin/fping

Modify the settings as needed; you will need to be root to execute these commands:

```
chown root.nagios /usr/local/groundwork/bin/fping
chmod 6755 /usr/local/groundwork/bin/fping
```

The configuration file gets installed as /usr/local/groundwork/etc/fping_process.conf, with nagios.nagios ownership and 600 permissions. Anything more permissive will cause the script to fail, for security reasons.

Now, copy the script to the /usr/local/groundwork/foundation/feeders directory (for use as a daemon) and to the /usr/local/groundwork/nagios/libexec directory (for use as a plugin) and make ownership/permission adjustments:

```
cp fping_process.pl /usr/local/groundwork/foundation/feeder/
chown nagios.nagios /usr/local/groundwork/foundation/feeder/fping_process.pl
chmod 755 /usr/local/groundwork/foundation/feeder/fping_process.pl
cp fping_process.pl /usr/local/groundwork/nagios/libexec/
chown nagios.nagios /usr/local/groundwork/nagios/libexec/fping_process.pl
chmod 755 /usr/local/groundwork/nagios/libexec/fping_process.pl
```

Last, if you wish to run the script as a persistent daemon rather than as a plugin, copy the supervise service directory to /usr/local/groundwork/services:

```
cp -R services/* /usr/local/groundwork/services
chown -R nagios.nagios /usr/local/groundwork/services/feeder-nagios-fping
```

The fping service should start within a second or two. Check that it is running by:

```
ps -ef | grep fping
```

You should see the fping_process.pl daemon running. It may already have an fping instance executing. If not, ensure that you have performed all the above steps correctly, and that all the files you installed are present and intact.

You may need to restart the services with the following sequence of commands:

```
/etc/init.d/httpd stop
/etc/init.d/snmpttd stop
/etc/init.d/gwservices restart
/etc/init.d/snmpttd start
/etc/init.d/httpd start
```

If you are sure that all is correct, and it still does not work, contact GroundWork support at support@groundworkopensource.comfor assistance, or use the GroundWork Support Forums at: http://www.groundworkopensource.com/community/forums/index.php

Tuning and Options

The following text describes a basic setup. Other variations are possible.

Now that passive host check results are being submitted to the command pipe, the Nagios configuration needs to be adjusted to properly make use of them.

The best way to do this is to set up a “passive” host profile. An example of a passive host profile is one where:

```
Active host checks are disabled
Passive host checks are enabled
Host freshness is being checked
```

There is no need to change the default host check command from a ping check, which is the normal default. You can leave the command as one that will work on the host, and simply turn off active checks of that host, by disabling them either in the template, for the host itself, or globally in the main Nagios configuration file. Enable passive host checks at the same time, however, or no results from the feeder will be posted. It is then unlikely that a host check will

ever be executed. If you do notice that a host check is coming in as “Active”, check the status of the fping feeder.

Setting a host freshness interval will allow the host to be actively checked, even if active checks are disabled, if no result from the fping feeder is received for this host within the timeout interval. The fping_process.pl script will run continuously, starting another cycle every 5 minutes if run by supervise, or every time it completes a run if run as a plugin. It will stage batches of hosts to avoid spiking the cpu with ping checks, and the command pipe with check results. It is unlikely that the fping feeder will fail, but leaving the active check in place provides insurance that hosts will be actively checked in case it does. The downside of setting a freshness check is that many blocking checks may be launched at once in the case that the fping feeder does not run. For this reason, GroundWork does not recommend setting a freshness threshold for host checks.

You can have a different indicator of the fping feeder function set up to alert you if it fails to submit a check. A service profile is included that you can import with the GroundWork Monitor Architect and apply to the localhost. It will check for the presence of the fping_process.pl command and, optionally and arguably a better test, it will accept passive check results from the fping feeder itself. See below. There is also an example of using the fping feeder as a plugin in this profile.

IMPORTANT: If you use this profile, choose the method by which you will run the fping process! You should not use BOTH the feeder and plugin methods on the same system, or you will be unnecessarily loading your system! Do not apply the fping_sweep service if you installed the feeder method. If you DO use the plugin method, apply it ONLY TO ONE host!

The fping feeder will also submit service check results. This function is on by default, but it can be turned off or on by editing the configuration file:

vi /usr/local/groundwork/etc/fping_process.conf

Modify the line:

send_service_check = no

to not send service checks, and set it back to:

send_service_check = yes

to send service checks.

Similarly, the host check submission can be disabled:

send_host_check = no

means disabled, and:

send_host_check = yes

means enabled (default).

Note that the service “icmp_ping” will be the only service that receives a passive check result by default. You may change this service name by editing the services_list entry in the configuration file, using a comma-delimited list (no spaces!). Be advised that this feeder will only submit results for services that exist in Monarch and are assigned to hosts.

In this release, you can no longer specify the services on the command line (e.g., in the services run file), as was possible under release 1.X of this software. All such modification must be done in the configuration file.

To use the supplied fping_feeder service profile and specifically the fping_receiver service it includes, simply enable service check results in the configuration file, and include the “fping_receiver” service in the configuration file services_list entry.

IMPORTANT: Once you have set up all the configuration options as you desire, you must edit this line in the configuration file:

enable_processing = no

and change it to:

enable_processing = yes

Until you make this change, no results will be returned by the script.

Feedback

Please submit your comments, suggestions, and improvements. We want your feedback!

Acknowledgments

Thanks to Peter Loh for his enhancements to the features and performance of this script, and to Thomas Stocking for earlier versions of this documentation. Thanks to Roland Schemers *(schemers@stanford.edu)* for fping, and to all the fping maintainers.