Nagios XI - NRPE Source-Based Install & Config

Nagios®

The Industry Standard in IT Infrastructure Monitoring

Purpose

This document describes how to install and configure NRPE from source for use with Nagios XI. The linux-nrpe-agent that ships with Nagios XI is only supported on a few select distributions (RHEL and CentOS). As most monitoring environments consist of many different distributions, you will find that you will need to compile nrpe and its associated plugins.

Target Audience

This document is intended for Nagios XI administrators who are new to NRPE or Nagios XI, and have to use a source-based install method of NRPE, usually due to unsupported Linux distributions or security restrictions in corporate build environments.

Downloading, Compiling, And Installing The NRPE Agent

On the remote host (client system) login as the root user, download the nrpe tarball (2.14 newest currently, though depending on when you use this document, there may be a newer version) and expand it:

```
cd /tmp
wget http://sourceforge.net/projects/nagios/files/nrpe-2.x/nrpe-2.14/nrpe-2.14.tar.gz
tar vfxz nrpe-2.14.tar.gz
```

Next we will compile it and install nrpe. The check_nrpe plugin will install into the default directory at: `/usr/local/nagios/libexec` To do so, run the following commands:

```
cd nrpe-2.14
./configure
make
make install
```

Copy over the nrpe.cfg from the sample-config directory into `/usr/local/nagios/etc`.

```
mkdir /usr/local/nagios/etc
cp sample-config/nrpe.cfg  /usr/local/nagios/etc
```

Configuring The Init System For NRPE

Now that nrpe has been installed, we need to set up an init script/system for the agent. You will first need to Install xinetd. Most Linux distributions include the package. As of the writing of this document, you can install nrpe on the following distributions, the following ways:

CentOS/RHEL/Fedora$ `yum install xinetd`
Ubuntu/Debian$ `apt-get install xinetd`
Gentoo$ `emerge -va xinetd`
ArchLinux$ `pacman -S xinetd`

Once xinetd is installed, we must configure it. The nrpe source has a pre-built xinetd config for the nrpe service. Change directory to the nrpe source and install it:

```
cd /tmp/nrpe-2.14
make install-xinetd
```

You should now have a config located at: `/etc/xinetd.d/nrpe`. This file contains an “only_from” directive which specifies which ip addresses are allowed to talk to the daemon. It is a space separated list of ip addresses. Add your Nagios XI server's IP to the directive:

```
nano /etc/xinetd.d/nrpe
```

Change:
```
only_from       = 127.0.0.1
```

To:
```
only_from       = 127.0.0.1 x.x.x.x
```
(where x.x.x.x is your Nagios XI server's ip)

Finally, we need to add the nrpe port and service declaration in /etc/services:

```
nano /etc/services
```

Add:
```
nrpe    5666/tcp
```

Let's test nrpe/xinetd to make sure it starts up cleanly:

```
service xinetd start
cd /usr/local/nagios/libexec
./check_nrpe -H localhost
```

You should see the following output:
```
NRPE v2.14
```

If you get the nrpe version number, the xinetd daemon was installed and configured correctly.

Downloading, Compiling, And Installing The Nagios Plugins

NRPE does not do much without plugins. So before we dive into configuration, download and build the Nagios-plugins (visit Nagios-Plugins.org/downloads to find the latest version of the plugins and adjust the version number in the commands accordingly):

```
/tmp
wget https://www.nagios-plugins.org/download/nagios-plugins-2.0.tar.gz
tar zxvf nagios-plugins-2.0.tar.gz
cd /tmp/nagios-plugins-2.0
./configure --with-nagios-user=nagios --with-nagios-group=nagios
make
make install
```

The nagios-plugins pack should now reside in `/usr/local/nagios/libexec`. The plugins are a mix of scripts and binaries. Take note of their location as this is where you will install additional plugins in the future.

To recap, nrpe and the nagios-plugins pack installed to `/usr/local/nagios/libexec`.
The nrpe.cfg config file is located at `/usr/local/nagios/etc`.

If you need to install to a different location due to internal policies, this can be done with with "**–prefix=$PREFIX**" directive. This is outside of the scope of this document but more information can be obtained through the Nagios documentation or directly from the Nagios forums. If you do use the --prefix option, you should use those paths instead of the ones mentioned and continue to use that directory throughout this document (for the remote client). Additionally, this guide assumes all checks will be statically defined in the nrpe.cfg file. If you need command-args enabled in your build, you should use "**./configure –enable-command-args**" when compiling. This will be covered in-depth in the advanced nrpe guide.

Configuration Overview

Essentially, there are three parts to a working nrpe agent:

1. NRPE and its configuration
2. A plugin with a configured command in nrpe.cfg
3. A check_nrpe service in Nagios XI

Parts one and two are both configured in the nrpe.cfg file, so we will begin there.

Configuration Of The Basic nrpe.cfg Options

NRPE runs as a daemon on the remote host. It accepts commands from a Nagios XI server over port 5666, executes them locally from `/usr/local/nagios/libexec`, and returns the check information to the Naigios XI server. These commands are specified in the nrpe.cfg and correspond to their respective plugins in libexec. This would be highly insecure if port 5666 was open to the world. Thus, there are a few directives in nrpe.cfg that must be edited for security reasons. Mainly the **`allowed_hosts`** directive. For the purposes of this document, we will use the command line text editor "nano", though feel free to use your editor of choice:

```
nano /usr/local/nagios/etc/nrpe.cfg
```

On line 81, you will find the allowed_hosts directive. This is a comma separated list of host IP addresses allowed to connect to the daemon on port 5666. **`Localhost (127.0.0.1)`** should already be listed and it is best to leave it as we will run nrpe plugins locally to check them before running them remotely from the Nagios XI server. Add a comma and then the IP address of your Nagios server with no space between them:

```
allowed_hosts=127.0.0.1,<your Nagios XI IP address>
```

The next directive of interest is **`dont_blame_nrpe=0`**. Funny name, but for a good reason. This directive allows command arguments to be passed to the nrpe daemon from the Nagios server. Whereas, by default, commands passed to nrpe have all their arguments/options set statically in nrpe.cfg. By changing the **0** to a **1**, nrpe will accept commands with arguments from the Nagios XI server. This setting is not without it's risks though, as it would open up the client to set up dynamic commands. **Depending on the level of security required in your environment, this directive may or may not be optional or viable**. An argument for switching this directive on is the amount of flexibility it gives the Nagios XI Admin to configure and reconfigure nrpe clients on the fly, all from within the Nagios XI interface. Additionally it allows for easier management of client nrpe configs across large monitoring environments, as the specific check can be almost entirely configured within Nagios XI, while clients share very similar, if not identical nrpe configs. If you do not have a specific security requirements or concerns, the **`dont_blame_nrpe`** directive is best set to "1".

The only other option of note (before we start on the command configuration) is the **`debug=0`** directive. If you experience problems with an nrpe setup, change the **0** to a **1** to enable debugging. It will be logged through the syslog facility by default and will provide verbose output from the nrpe daemon. This log can be reached "in most cases" by entering the following from the command line:

```
tail -f /var/log/syslog
```

Test And Verify The NRPE Installation

NRPE uses port 5666 and runs as a daemon. After installation, verify that the process is running on the remote host and port 5666 is reachable from the Nagios XI server. On the remote host:

```
service xinetd status
```

Your output should look something like:

```
       xinetd (pid  1150) is running...
```

If the process is not running, start it with the following command:

```
service xinetd start
```

We will use nmap to make sure the port is open. If you do not currently have nmap installed, run the following from the command line on your Nagios XI server (for CentOS/RHEL, for other distributions consult your package repos):

```
yum install nmap
```

Log onto the Nagios XI server in a terminal and run the following command (where x.x.x.x is the IP address of the remote host):

```
nmap x.x.x.x -p 5666
```

If the port is closed, you may have a router/firewall between the Nagios XI server and host, or port 5666 may not be open on the remote host. As nrpe loads through xinetd, the port should be opened automatically. If you installed nrpe and used an init script loading method instead, you will have to open port 5666 by hand with iptables or whatever the utility your remote host uses for port management.

If nmap reports the port is open, you are ready to test nrpe to the remote host. You can run nrpe without any additional commands to check the remote nrpe version, effectively testing the status of the agent. From the Nagios XI server command line run the following commands (where x.x.x.x is the IP address of the remote host):

```
cd /usr/local/nagios/libexec
./check_nrpe -H x.x.x.x
```

You should receive output resembling:

```
NRPE v2.14
```

If you get the remote host's nrpe version as output, then your installation is working and Nagios XI is all set to begin checking the remote host through the nrpe agent.

Configuration of Static Command Directives in nrpe.cfg

On your remote host the command directives at the bottom of the nrpe.cfg file are aliases to plugins. These are the plugins that are executed on the remote system when it receives a check command through nrpe from the Nagios XI sever. You have to configure the command to match the plugin's location and arguments you wish to pass to the plugin. In a static configuration, these directives include the entirety of the command: alias, path, plugin and arguments. The commands are in the format:

```
command[alias]=/path/plugin arguments
```

An actual command directive would look like:

```
command[check_total_procs]=/usr/local/nagios/libexec/check_procs -w 150 -c 200
```

For this example, we give the command the name/alias **`[check_total_procs]`**. Everything after the "=" is the actual command run on the client. We give the command a full path: `/usr/local/nagios/libexec/check_procs` where "check_procs" is the binary/script plugin in the /usr/local/nagios/libexec folder and "-w 150 -c 200" are the command arguments for the plugin. -w 150 specifies the **warning** level, which is the point at which Nagios XI will report a WARNING once the client reaches 150 processes. The -c 200 is the critical setting, specifying that the service will report CRITICAL in Nagios XI once the client reaches 200 processes. These thresholds are user configurable, and tend to be unique in syntax and in scope of any given plugin.

Writing, Testing, And Deploying Your First Custom Command Directive

The nrpe.cfg file should have some default commands already setup, we will be editing them to learn about nrpe and the default plugins. The process for adding a new command directive (at least when just starting out) will include creating the command with your desired options and then testing it from the command line on the remote host. Only after you have attempted to run the check from the remote host should you edit the nrpe.cfg file. Then only after successfully runing the check from the Nagios XI server command line using nrpe, should you create the associated check in Nagios XI. Using this process will greatly reduce the troubleshooting time necessary when you are learning nrpe and Nagios XI. Additionally, it will make support requests easier and faster to resolve.

You were introduced to the check_procs plugin above, now it is time to write a more specialized version of the command. Most plugins have help/usage information. On the remote host, navigate to the default plugin directory and run the plugin with “-h” or rarely, “--help”:

```
cd /usr/local/nagios/libexec
./check_procs -h
```

This procedure can and should be repeated for most plugins before deployment. Looking at the check_procs help, you will notice many options that are not used by the default nrpe.cfg check. For this example, we will check for specific process (cron) by string (“-a” switch

for check_procs). Run a test from the command line, starting with just a check for the process on the remote host:

```
cd /usr/local/nagios/libexec
./check_procs -a cron
```

You should get output resembling:

```
PROCS OK: 2 processes with args 'cron'
```

The check works locally. Now let's test it through nrpe. First, we need to write the new command to the remote host's nrpe.cfg. On the remote host:

```
nano /usr/local/nagios/etc/nrpe.cfg
```

Add a new command at the end of the file, naming the check "check_procs_cron" and adding the cron string:

```
command[check_procs_cron]=/usr/local/nagios/libexec/check_procs -a cron
```

Save the file out and restart nrpe:

```
service xinetd restart
```

On the Nagios XI server, lets run a test from the command line (where x.x.x.x is the ip of the remote host) before we define the check in Nagios XI:

```
cd /usr/local/nagios/libexec
./check_nrpe -H x.x.x.x -c check_procs_cron
```

You should, once again, receive output resembling:

```
PROCS OK: 1 process with args 'cron'
```

If so, the remote host is configured and all that is left is to add the check to Nagios XI. If not, review the previous steps for syntax errors and try the plugin locally once again before proceeding to this next step.

Configuration Of A Nagios XI NRPE Service Check Through The Core Config Manager

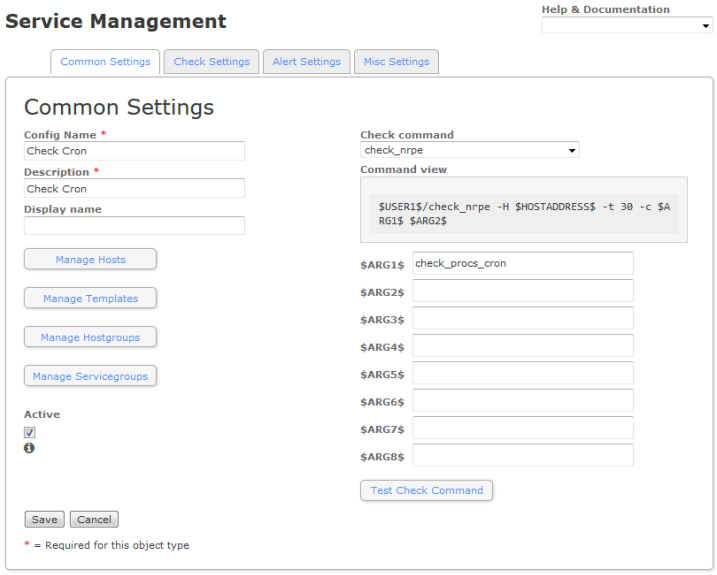
The final thing to do for this custom check is to configure it in Nagios XI through the Core Config Manager (CCM). Log into the Nagios XI server through the web interface.

Navigate to **Configure → Core Config Manager → Services → Add New**. Specify the configuration name and description for the check, we will use "Check Cron".

Next select *check_nrpe* in the **Check_command** drop down list. Now set up the command arguments under **Command view**. **$ARG1$** is the check you wish to perform, in this case enter: check_procs_cron. **$ARG2$** is used for extra command arguments, but as they are setup on the remote host in it's nrpe.cfg file **(-a cron)**, we can leave the field blank.

The check needs to be applied to a host, so click the **Manage Hosts** button. Select the remote host from the list and click **Add Selected.** You should see the host appear in the right hand panel under **Assigned**. Click **Close**.

Note: If you do not see the remote host in the list, you need to create a new host definition in Nagios XI before



you can proceed.
One thing to note is that you can add multiple hosts to one service check, effectively checking multiple hosts with the same check configuration, though you will have to setup nrpe and the check on each individual remote host.

The last thing to do in the Nagios XI configuration is to set up a few defaults that Nagios XI expects. Click the **Check Settings** tab. At minimum, we need to setup check intervals, attempts, and a period. **Check interval** specifies how often the check is run. **Retry interval** specifies the time between check retries when the service check has failed (SOFT STATE). **Max check attempts** specifies the number of retries a check will attempt before it is marked as a HARD STATE fail. The last required setting to set on this tab is the **Check period**. This specifies the time period the check should run and can be configured for certain days and time frames. *xi_timeperiod_24x7* will be fine for this example.

Last, click the **Alert Settings** and set the **Notification period** to *xi_timeperiod_24x7*. This specifies the time period for notifications. Click **Manage Contacts** and add a contact to the check if you want.

Click **Save** and **Apply Configuration.**

Now we want to verify the check is running so navigate to **Home** → **Service Detail**. It may take a minute for the service to change from Pending to a different state.

192.168.4.250	/ Disk Usage	Ok	3h 13m 59s	1/5	2015-09-03 14:21:34	DISK OK - free space: / 12499 MB (87% inode=99%):
	Check Cron	Ok	21s	1/3	2015-09-03 14:25:09	crond.service - Command Scheduler

You can add more hosts to the check by editing the service check and adding hosts or hostgroups. This procedure can be used as an example/abstract for setting up different nrpe service checks.

Finishing Up

If you have any issues with configuring nrpe service checks within your Nagios XI system, please post your questions on the Nagios Support Forums at the following URL:

http://support.nagios.com/