



NRPE DOCUMENTATION

Copyright (c) 1999-2017 Ethan Galstad

Last Updated: 17 November 2017

Contents

1. Introduction.....	2
a) Purpose.....	2
b) Design Overview.....	2
2. Example Uses.....	3
a) Direct Checks.....	3
b) Indirect Checks.....	3
3. Installation.....	4
a) Prerequisites.....	4
b) Remote Host Setup.....	4
c) Monitoring Host Setup.....	8
4. Customizing Your Configuration.....	13
a) Remote Host Configuration.....	13
b) Monitoring Host Configuration.....	14
5. Upgrading.....	15
a) Monitoring Host Upgrade.....	15
b) Remote Host Upgrades.....	16
6. Troubleshooting.....	17

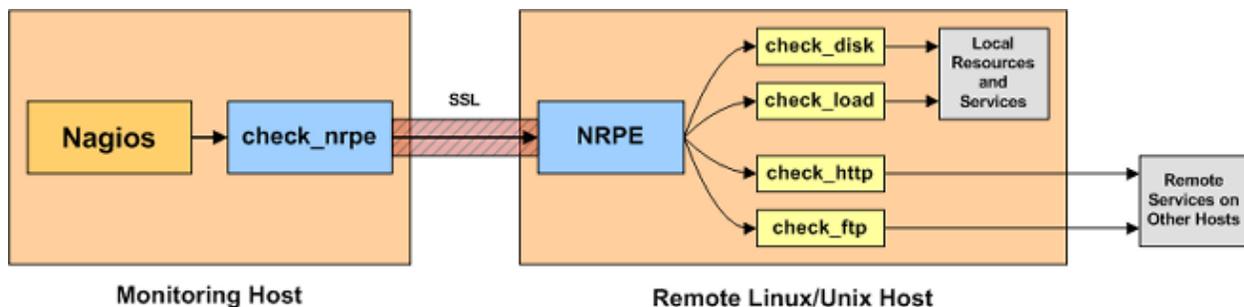
1. INTRODUCTION

a) Purpose

The NRPE addon is designed to allow you to execute Nagios plugins on remote Linux/Unix machines. The main reason for doing this is to allow Nagios to monitor "local" resources (like CPU load, memory usage, etc.) on remote machines. Since these public resources are not usually exposed to external machines, an agent like NRPE must be installed on the remote Linux/Unix machines.

Note: It is possible to execute Nagios plugins on remote Linux/Unix machines through SSH. There is a `check_by_ssh` plugin that allows you to do this. Using SSH is more secure than the NRPE addon, but it also imposes a larger (CPU) overhead on both the monitoring and remote machines. This can become an issue when you start monitoring hundreds or thousands of machines. Many Nagios admins opt for using the NRPE addon because of the lower load it imposes.

b) Design Overview



The NRPE addon consists of two pieces:

- The `check_nrpe` plugin, which resides on the local monitoring machine
- The `NRPE` daemon, which runs on the remote Linux/Unix machine

When Nagios needs to monitor a resource or service from a remote Linux/Unix machine:

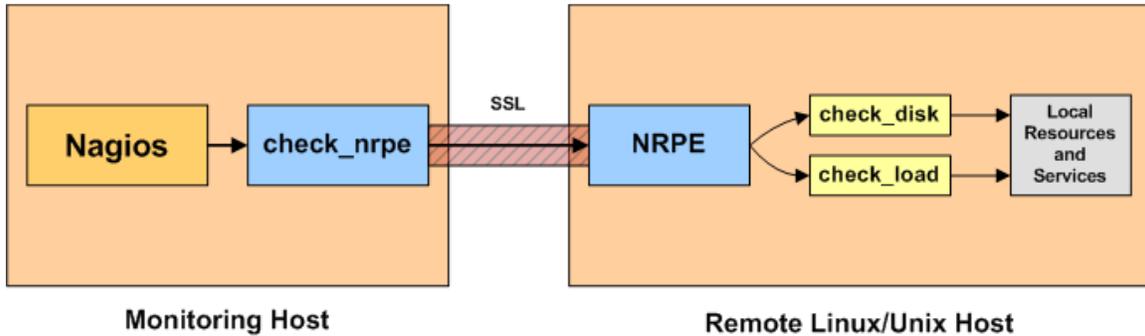
- Nagios will execute the `check_nrpe` plugin and tell it what service needs to be checked
- The `check_nrpe` plugin contacts the `NRPE` daemon on the remote host over an (optionally) SSL protected connection
- The `NRPE` daemon runs the appropriate Nagios plugin to check the service or resource
- The results from the service check are passed from the `NRPE` daemon back to the `check_nrpe` plugin, which then returns the check results to the Nagios process.

Note: The `NRPE` daemon requires that Nagios plugins be installed on the remote Linux/Unix host. Without these, the daemon wouldn't be able to monitor anything.

2. EXAMPLE USES

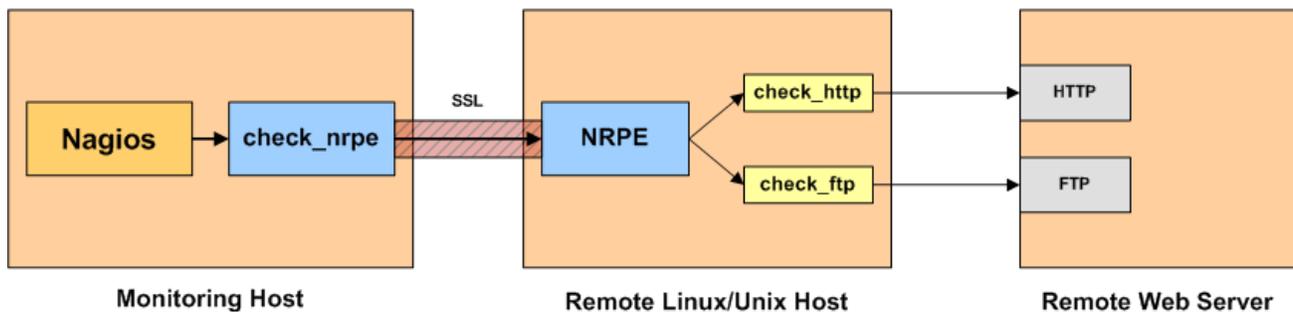
a) Direct Checks

The most straight forward use of the NRPE addon is to monitor "local" or "private" resources on a remote Linux/Unix Machine. This includes things like CPU load, memory usage, swap usage, current users, disk usage, process states, etc.



b) Indirect Checks

You can also use the NRPE addon to indirectly check "public" services and resources of remote servers that might not be reachable directly from the monitoring host. For instance, if the remote host that the NRPE daemon and plugins are installed on can talk to the a remote web server (but the monitoring host cannot), you can configure the NRPE daemon to allow you to monitor the remote web server indirectly. The NRPE daemon is essentially acting as proxy in this case.



3. INSTALLATION

In order to use the NRPE add-on, you'll need to perform some tasks on both the monitoring host and the remote Linux/Unix host that the NRPE daemon is installed on. I'll cover both of these tasks separately.

You need to decide if you will have the NRPE daemon running at all times, or if it will start for each incoming connection (ex. Under inetd or xinetd).

Note: As of version 3.0, NRPE has been updated to make building and installing much easier on a wide variety of operating systems. The instructions presented here are based on hosts running a generic common Linux distribution such as CentOS, Fedora or SUSE. When naming conventions, commands, etc. vary across different Linux distributions and UNIX variants, I will usually note the differences. But the instructions provided here may have to be altered a bit for your situation.

a) Prerequisites

In order to complete these installation instructions, you'll need:

- Root access on the remote Linux/Unix host
- Access to the nagios user account on the monitoring host
- Access to the source, either via Internet, local network, disk, or CD-ROM
- Development tools: compiler and linker (gcc preferred)
- SSL headers and libraries

b) Remote Host Setup

i. Account Setup

As of NRPE version 3.0, the Makefile includes targets to add the required users and groups to the computer's local accounts (usually `/etc/passwd`), if necessary. If you will be adding them to LDAP or some other authentication system, you will have to do it yourself.

ii. Install the Nagios Plugins

Create a directory for storing the downloads, if you don't already have one.

```
# mkdir ~/downloads
# cd ~/downloads
```

Download the source code tarball of the Nagios plugins (visit <http://www.nagios.org/downloads/> for links to the latest versions). At the time of writing, the latest stable version of the Nagios plugins was 2.1.1.

```
# wget http://nagios-plugins.org/download/nagios-plugins-2.2.1.tar.gz
```

Extract the Nagios plugins source code tarball.

```
# tar xzf nagios-plugins-2.2.1.tar.gz
# cd nagios-plugins-2.2.1
```

Note: on some systems, you will have to run the extract this way:

```
# gunzip -c nagios-plugins-2.2.1.tar.gz | tar xf -
```

Compile and install the plugins.

```
# ./configure
# make
# make install
```

Depending on the version of the plugins, the permissions on the plugin directory and the plugins may need to be fixed at this point. If so run the following commands:

```
# useradd nagios
# groupadd nagios
# usermod -a -G nagios nagios
# chown nagios.nagios /usr/local/nagios
# chown -R nagios.nagios /usr/local/nagios/libexec
```

iii. Install xinetd

If you will be running NRPE per-connections, some distributions, such as Fedora Core 6, don't ship with xinetd installed by default, so install it with the following command:

```
# yum install xinetd
```

iv. Install the NRPE daemon

Download the source code tarball of the NRPE addon (visit <https://www.nagios.org/downloads/nagios-core-addons/> for links to the latest versions). At the time of writing, the latest version of NRPE was 3.2.1.

```
# cd ~/downloads
# wget https://github.com/NagiosEnterprises/nrpe/releases/download/nrpe-3.2.1/nrpe-3.2.1.tar.gz
```

Extract the NRPE source code tarball:

```
tar xzf nrpe-3.2.1.tar.gz
cd nrpe-nrpe-3.2.1
```

Compile the NRPE addon:

```
# ./configure
# make all
```

If you didn't create the groups and users in (i) above, do it now:

```
# make install-groups-users
```

Install the NRPE plugin (for testing), daemon, and sample daemon configuration file.

```
# make install
# make install-config
```

If you want NRPE to run per-connection under inetd, xinetd, launchd, systemd, smf, etc. run the following command:

```
# make install-inetd
```

Make sure `nrpe 5666/tcp` is in your `/etc/services` file, if applicable.

If you want to run NRPE all the time under init, launchd, systemd, smf, etc. run the following command:

```
# make install-init
```

You may need to reload or restart the controlling daemon using one of the following (or similar) commands:

```
# service xinetd restart
# systemctl reload xinetd           # systemd
# systemctl enable nrpe && systemctl start nrpe # systemd
# svcadm enable nrpe                 # smf
# initctl reload-configuration       # upstart
```

v. Test the NRPE daemon locally

Its time to see if things are working properly. Make sure the nrpe daemon is running:

```
# netstat -at | egrep "nrpe|5666"
tcp    0      0  *:nrpe  ::*    LISTEN
```

If the second line above shows up, great! If it doesn't, make sure of the following:

- You added the nrpe entry to your `/etc/services` file
- The `only_from` directive in the `/etc/xinetd.d/nrpe` file contains an entry for "127.0.0.1"
- xinetd is installed and started
- The appropriate line in `/etc/inetd.conf` has been uncommented
- Check the system log files for references about xinetd or nrpe and fix any problems that are reported

Next, check to make sure the NRPE daemon is functioning properly. To do this, run the `check_nrpe` plugin that was installed for testing purposes. You should see the second line below:

```
# /usr/local/nagios/libexec/check_nrpe -H localhost
NRPE v3.2.1
```

If everything worked, add the hostname or IP address of the nagios server to the `/etc/xinetd.d/nrpe` file, or `/etc/hosts-allow` and `hosts-deny`.

vi. Open firewall rules

If the server has a firewall running, you need to allow access to the NRPE port (5666) from the Nagios server.

In Fedora and Red Hat Linux, you would use the following commands:

```
# iptables -I RH-Firewall-1-INPUT -p tcp -m tcp --dport 5666 -j ACCEPT
# service iptables save
```

On other systems and other firewalls, check the documentation or have an administrator open the port.

vii. Customize the NRPE commands

The sample NRPE config file that got installed contains several command definitions that you'll likely use to monitor this machine. The command definitions are used to (surprise) define commands that the NRPE daemon will run to monitor local resources and services. The sample command definitions run some of the plugins that were installed in step 2. You can edit the command definitions, add new commands, etc, by editing the NRPE config file using your favorite editor:

```
# vimacs /usr/local/nagios/etc/nrpe.cfg
```

More information on customizing the commands can be found on page 16 in the section titled "Customizing Your Configuration".

For the time being, I'll assume you're using the sample commands that are defined. You can test some of these by running the following commands:

```
/usr/local/nagios/libexec/check_nrpe -H localhost -c check_users
```

```
/usr/local/nagios/libexec/check_nrpe -H localhost -c check_load  
/usr/local/nagios/libexec/check_nrpe -H localhost -c check_hda1  
/usr/local/nagios/libexec/check_nrpe -H localhost -c check_total_procs  
/usr/local/nagios/libexec/check_nrpe -H localhost -c check_zombie_procs
```

At this point, you are done installing and configuring NRPE on the remote host. Now its time to install a component and make some configuration entries on your monitoring server...

c) Monitoring Host Setup

On the monitoring host (the machine that runs Nagios), you'll need to do just a few things:

- Install the `check_nrpe` plugin
- Create a Nagios command definition for using the `check_nrpe` plugin
- Create Nagios host and service definitions for monitoring the remote host

These instructions assume that you have already installed Nagios on this machine according to the quickstart installation guide. The configuration examples that are given reference templates that are defined in the sample `localhost.cfg` and `commands.cfg` files that get installed if you follow the quickstart.

i. Install the `check_nrpe` Plugin

Download the source code tarball of the NRPE addon (visit <https://www.nagios.org/downloads/nagios-core-addons/> for links to the latest versions). At the time of writing, the latest version of NRPE was 3.2.1.

```
# cd ~/downloads
# wget https://github.com/NagiosEnterprises/nrpe/releases/download/nrpe-3.2.1/nrpe-3.2.1.tar.gz
```

Extract the NRPE source code tarball:

```
# tar xzf nrpe-3.2.1.tar.gz
# cd nrpe-nrpe-3.2.1
```

Compile the NRPE addon:

```
# ./configure
# make check_nrpe
```

Install the NRPE plugin.

```
# make install-plugin
```

ii. Test communication with the NRPE daemon

Make sure the `check_nrpe` plugin can talk to the NRPE daemon on the remote host. Replace "192.168.0.1" in the command below with the IP address of the remote host that has NRPE installed. You should get back the second line below:

```
/usr/local/nagios/libexec/check_nrpe -H 192.168.0.1
NRPE v3.2.1
```

If the plugin returns a timeout error, check the following:

- Make sure there isn't a firewall between the remote host and the monitoring server that is blocking communication
- Make sure that the NRPE daemon is installed properly and running on the remote host
- Make sure the remote host doesn't have local firewall rules that prevent the monitoring server from talking to the NRPE daemon

iii. Create a command definition

You'll need to create a command definition in one of your Nagios object configuration files in order to use the `check_nrpe` plugin. Open the sample `commands.cfg` file for editing...

```
# vimacs /usr/local/nagios/etc/commands.cfg
```

and add the following definition to the file:

```
define command{
    command_name    check_nrpe
    command_line    $USER1$/check_nrpe -H $HOSTADDRESS$ -c $ARG1$
}
```

You are now ready to start adding services that should be monitored on the remote machine to the Nagios configuration...

iv. Create host and service definitions

You'll need to create some object definitions in order to monitor the remote Linux/Unix machine. These definitions can be placed in their own file or added to an already existing object configuration file.

First, its best practice to create a new template for each different type of host you'll be monitoring. Let's create a new template for linux boxes.

```
define host{
    name          linux-box          ; Name of this template
    use           generic-host       ; Inherit default values
    check_period  24x7
    check_interval 5
    retry_interval 1
    max_check_attempts 10
    check_command check-host-alive
    notification_period 24x7
    notification_interval 30
    notification_options d,r
    contact_groups admins
    register      0 ; DONT REGISTER THIS - ITS A TEMPLATE
}
```

Notice that the linux-box template definition is inheriting default values from the generic-host template, which is defined in the sample localhost.cfg file that gets installed when you follow the Nagios quickstart installation guide.

Next, define a new host for the remote Linux/Unix box that references the newly created linux-box host template.

```
define host{
    use          linux-box          ; Inherit default values from a template
    host_name    remotehost        ; The name we're giving to this
server
    alias        Fedora Core 6     ; A longer name for the server
    address      192.168.0.1       ; IP address of the server
}
```

Next, define some services for monitoring the remote Linux/Unix box. These example service definitions will use the sample commands that have been defined in the `nrpe.cfg` file on the remote host.

The following service will monitor the CPU load on the remote host. The "check_load" argument that is passed to the `check_nrpe` command definition tells the NRPE daemon to run the "check_load" command as defined in the `nrpe.cfg` file.

```
define service{
    use          generic-service
    host_name    remotehost
    service_description    CPU Load
    check_command    check_nrpe!check_load
}
```

The following service will monitor the number of currently logged in users on the remote host.

```
define service{
    use          generic-service
    host_name    remotehost
    service_description    Current Users
    check_command    check_nrpe!check_users
}
```

The following service will monitor the free drive space on /dev/hda1 on the remote host.

```
define service{
    use          generic-service
    host_name    remotehost
    service_description    /dev/hda1 Free Space
    check_command    check_nrpe!check_hda1
}
```

The following service will monitor the total number of processes on the remote host.

```
define service{
    use          generic-service
    host_name    remotehost
    service_description    Total Processes
    check_command    check_nrpe!check_total_procs
}
```

The following service will monitor the number of zombie processes on the remote host.

```
define service{
    use          generic-service
    host_name    remotehost
    service_description    Zombie Processes
    check_command    check_nrpe!check_zombie_procs
}
```

Those are the basic service definitions for monitoring the remote host. If you would like to add additional services to be monitored, read the "Customizing Your Configuration" section starting on page 16.

v. Restart Nagios

At this point you've installed the check_nrpe plugin and addon host and service definitions for monitoring the remote Linux/Unix machine. Now its time to make those changes live...

Verify your Nagios configuration files.

```
# /usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg
```

If there are errors, fix them. If everything is fine, restart Nagios using one of the commands below, or whatever is appropriate on your server.

```
# service nagios restart
# systemctl restart nagios           # systemd
# svcadm disable nrpe && svcadm enable nrpe   # smf
# /etc/rc.d/nrpe restart              # various SysV
```

That's it! You should see the host and service definitions you created in the Nagios web interface. In a few minutes Nagios should have the current status information for the remote Linux/Unix machine.

Since you might want to monitor more services on the remote machine, I would suggest you read the next section as well. :-)

Also, when it comes time to upgrade the version of NRPE you're running, its pretty easy to do. The initial installation was the toughest, but upgrading is a snap.

4. CUSTOMIZING YOUR CONFIGURATION

You'll might want to monitor more services on the remote Linux/Unix box than those that are described in this documentation. This is easy to accomplish.

Anytime you want to monitor a new service on a remote host using the NRPE addon, you have to do two things:

1. Add a new command definition to the `nrpe.cfg` file on the remote host
2. Add a new service definition to your Nagios configuration on the monitoring host

Let's say you want to monitor the swap usage on the remote Linux/Unix host. Here are the steps you'd need to follow...

a) Remote Host Configuration

You can use the `check_swap` plugin to monitor swap usage on the machine. Assuming you followed the installation instructions in this document, the `check_swap` plugin should already be installed in `/usr/local/nagios/libexec`.

Login as the `nagios` user on the remote host.

Run the `check_swap` plugin manually and tweak the command line options to specify the desired warning and critical free swap space thresholds. Make sure the full command line returns the expected output you want from the plugin. For this example, let's say you want a critical alert if swap free space is less than 10% and a warning if free space is less than 20%. Here's the command line that would accomplish that:

```
# /usr/local/nagios/libexec/check_swap -w 20% -c 10%
```

Now that you know the command line that should be execute, open the NRPE configuration file.

```
$ vimacs /usr/local/nagios/etc/nrpe.cfg
```

Add a new `check_swap` command definition that uses the command line from above and save the file.

```
command[check_swap]=/usr/local/nagios/libexec/check_swap -w 20% -c 10%
```

If you're running the NRPE daemon as a standalone daemon you'll need to restart it. If you're running it under the `inetd/xinetd` superserver you don't need to do anything more.

b) Monitoring Host Configuration

On the monitoring host, you need to define a new service for monitoring the swap usage on the remote host. Add the following entry to one of your object configuration files.

```
define service{
    use          generic-service
    host_name    remotehost
    service_description    Swap Usage
    check_command    check_nrpe!check_swap
}
```

Notice that the check commands is passing "check_swap" to the check_nrpe command definition. This will cause the NRPE daemon to run the check_swap command that was defined in the nrpe.cfg file on the remote host in the previous step.

Next, verify your Nagios configuration files and restart Nagios.

```
# /usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg
# service nagios restart
```

That's it! You are now monitoring a new service on the remote host using the NRPE addon.

5. UPGRADING

At some point in time you'll want to upgrade the version of the NRPE addon that you're running. The upgrade process is fairly pain free. Here are the details...

a) Monitoring Host Upgrade

On the monitoring host (the machine that runs Nagios), you only have to upgrade the `check_nrpe` plugin. Here's how...

Login as the nagios user and create a directory for storing the downloads.

```
$ mkdir ~/downloads
$ cd ~/downloads
```

Download the source code tarball of the NRPE addon (visit <https://www.nagios.org/downloads/nagios-core-addons/> for links to the latest versions). At the time of writing, the latest version of NRPE was 3.2.1.

```
$ cd ~/downloads
$ wget https://github.com/NagiosEnterprises/nrpe/releases/download/nrpe-3.2.1/nrpe-3.2.1.tar.gz
```

Extract the NRPE source code tarball:

```
$ tar xzf nrpe-3.2.1.tar.gz
$ cd nrpe-nrpe-3.2.1
```

Compile the NRPE addon:

```
$ ./configure
$ make check_nrpe
```

Install the NRPE plugin.

```
$ make install-plugin
```

You're done on the monitoring server!

b) Remote Host Upgrades

You'll need to upgrade the NRPE daemons on all remote hosts that are being monitored using the NRPE addon. Here's how...

Login as the nagios user and create a directory for storing the downloads.

```
$ mkdir ~/downloads
$ cd ~/downloads
```

Download the source code tarball of the NRPE addon (visit <https://www.nagios.org/downloads/nagios-core-addons/> for links to the latest versions). At the time of writing, the latest version of NRPE was 3.2.1.

```
$ cd ~/downloads
$ wget https://github.com/NagiosEnterprises/nrpe/releases/download/nrpe-3.2.1/nrpe-3.2.1.tar.gz
```

Extract the NRPE source code tarball:

```
$ tar xzf nrpe-3.2.1.tar.gz
$ cd nrpe-nrpe-3.2.1
```

Compile the NRPE addon:

```
$ ./configure
$ make all
```

Install the NRPE daemon.

```
$ make install-daemon
```

If you're running the NRPE daemon as a standalone daemon, first kill the old daemon process and then start the (new) daemon up again.

You're done with the upgrade on the remote host!

6. TROUBLESHOOTING

Here are some tips for troubleshooting some of the more common errors with the NRPE addon. If you encounter problems that aren't covered here, post a message on the support forum at

<https://support.nagios.com/forum/>

The check_nrpe plugin returns "CHECK_NRPE: Socket timeout after 10 seconds" or "Connection refused or timed out"

This error can indicate several things:

- The command that the NRPE daemon was asked to run took longer than 10 seconds to execute. This is the most likely cause if the error message was "CHECK_NRPE: Socket timeout after 10 seconds". Use the `-t` command line option to specify a longer timeout for the `check_nrpe` plugin. The following example will increase the timeout to 30 seconds:

```
/usr/local/nagios/check_nrpe -H localhost -c somecommand -t 30
```

- The NRPE daemon is not installed or running on the remote host. Verify that the NRPE daemon is running as a standalone daemon or under `inetd/xinetd` with one of the following commands:

```
$ ps axuw | grep nrpe
$ netstat -at | grep nrpe
```

- There is a firewall that is blocking the communication between the monitoring host (which runs the `check_nrpe` plugin) and the remote host (which runs the NRPE daemon). Verify that the firewall rules (e.g. `iptables`) that are running on the remote host allow for communication and make sure there isn't a physical firewall that is located between the monitoring host and the remote host.

The check_nrpe plugin returns "CHECK_NRPE: Received 0 bytes from daemon. Check the remote server logs for an error message."

First thing you should do is check the remote server logs for an error message. Seriously. :-) This error could be due to the following problem:

- The `check_nrpe` plugin was unable to complete an SSL handshake with the NRPE daemon. An error message in the logs should indicate whether or not this was the case. Check the versions of OpenSSL that are installed on the monitoring host and remote host. If you're running a commercial version of SSL on the remote host, there might be some compatibility problems.

The check_nrpe plugin returns "NRPE: Unable to read output"

This error indicates that the command that was run by the NRPE daemon did not return any character output. This could be an indication of the following problems:

- An incorrectly defined command line in the command definition. Verify that the command definition in your NRPE configuration file is correct.
- The plugin that is specified in the command line is malfunctioning. Run the command line manually to make sure the plugin returns some kind of text output.

The check_nrpe plugin returns "NRPE: Command 'x' not defined"

This error means that you didn't define command x in the NRPE configuration file on the remote host. On the remote host, add the command definition for x. See the existing command definitions in the NRPE configuration file for more information on doing this. If you're running the NRPE daemon as a standalone daemon (and not under inetd or xinetd), you'll need to restart it in order for the new command to be recognized.

The check_nrpe plugin returns "NRPE: Command timed out after x seconds"

This error indicates that the command that was run by the NRPE daemon did not finish executing within the specified time. You can increase the timeout for commands by editing the NRPE configuration file and changing the value of the command_timeout variable. If you're running the NRPE daemon as a standalone daemon (and not under inetd or xinetd), you'll need to restart it in order for the new timeout to be recognized.

How to go about debugging other problems...

When debugging problems it may be useful to edit the NRPE configuration file and change the debug=0 entry to debug=1. Once you do that, restart the NRPE daemon if it is running as a standalone daemon. After you try using the check_nrpe plugin again, you should be able to see some debugging information in the log files of the remote host. Check your logs carefully - they should be able to help provide clues as to where the problem lies...