

NRPEListener.dll

This module accepts incoming NRPE connections and responds by executing various checks and returns their result. To use this you need to have check_nrpe or another NRPE client. This is similar to check_nt (NSClient) but much more flexible and supports encryption. This only drawback is that it lacks any authorization.

As this module has the ability to generate command handlers by configuration there are command handlers but nothing built in. This is present for compatibility only it is suggested to use the [\[CheckExternalScripts\]](#) instead.

Configuration Sections

NRPE Section

This is a wrapper page the actual data is on the following page [NRPEListener/config/nrpe](#)

1. 1. 1. 1. Overview
 1. port
 2. allowed hosts
 3. use_ssl
 4. bind to address
 5. command timeout
 6. allow arguments
 7. allow nasty meta chars
 8. socket timeout
 9. script dir
 10. performance data
 11. socket back log
 12. string length

Overview

This is configuration for the [NRPE module](#) that controls how the [NRPE listener](#) operates.

Option	Default	Description
port	5666	The port to listen to
allowed_hosts		A list of hosts allowed to connect via NRPE.
use_ssl	1	Boolean value to toggle SSL encryption on the socket connection
command_timeout	60	The maximum time in seconds that a command can execute. (if more then this execution will be aborted). NOTICE this only affects external commands not internal ones.
allow_arguments	0	A Boolean flag to determine if arguments are accepted on the incoming socket. If arguments are not accepted you can still use external commands that need arguments but you have to define them in the NRPE handlers below. This is similar to the NRPE "dont_blame_nrpe" option.
allow_nasty_meta_chars	0	Allow NRPE execution to have ?nasty? meta characters that might affect execution of external commands (things like > ? etc).

socket_timeout	30	The timeout when reading packets on incoming sockets. If the data has not arrived within this time we will bail out. and discard the connection.
----------------	----	--

Advanced options:

Option	Default	Description
performance_data	1	Send performance data back to nagios (set this to 0 to remove all performance data)
socket_back_log		Number of sockets to queue before starting to refuse new incoming connections. This can be used to tweak the amount of simultaneous sockets that the server accepts. This is an advanced option and should not be used.
string_length	1024	Length of payload to/from the NRPE agent. This is a hard specific value so you have to "configure" (read recompile) your NRPE agent to use the same value for it to work.
script_dir		Load all scripts in a directory and use them as commands. Probably dangerous but usefull if you have loads of scripts :)
bind_to_address		The address to bind to when listening to sockets.

port

The port to listen to

Default

5666

allowed_hosts

A list (comma separated) with hosts that are allowed to poll information from NRPE. This will replace the one found under Setting for NRPE if present. If not present the same option found under Settings will be used. If both are blank all hosts will be allowed to access the system

Default

Empty list (falls back to the one defined under [Settings])

use_ssl

Boolean value to toggle SSL (Secure Socket Layer) encryption on the socket connection. This corresponds to the -n flag in check_nrpe

Values

Value	Meaning
0	Don't use SSL
1	Use SSL encryption

Default

1 (enabled)

bind_to_address

The address to bind to when listening to sockets. If not specified the "first" (all?) one will be used (often the correct one).

Values

IP address of any interface of the server.

Default

Empty (first (all?) interface will be used)

command_timeout

The maximum time in seconds that a command can execute. (if more then this execution will be aborted). NOTICE this only affects external commands not internal ones so internal commands may execute forever.

It is usually a good idea to set this to less then the timeout used with check_nrpe

Default

60

allow_arguments

A Boolean flag to determine if arguments are accepted on the incoming socket. If arguments are not accepted you can still use external commands that need arguments but you have to define them in the NRPE handlers below. This is similar to the NRPE "dont_blame_nrpe" option.

NOTICE That there are more then one place to set this!

Default

0 (means don't allow arguments)

Values

Value	Meaning
0	Don't allow arguments
1	Allow arguments.

allow_nasty_meta_chars

Allow NRPE execution to have ?nasty? meta characters that might affect execution of external commands (things like > ? etc).

Default

0 (means don't allow meta characters)

Values

Value	Meaning
0	Don't allow meta characters
1	Allow meta characters

socket_timeout

The timeout when reading packets on incoming sockets. If the data has not arrived within this time we will bail out. and discard the connection.

Default

30 seconds

script_dir

Load all scripts in a directory and use them as commands. Probably dangerous but useful if you have loads of scripts :)

Default

Empty (don't load any scripts)

performance_data

Send performance data back to Nagios (set this to 0 to remove all performance data)

Default

1

Values

Value	Meaning
0	Don't send performance data
1	Send performance data

socket_back_log

Number of sockets to queue before starting to refuse new incoming connections. This can be used to tweak the amount of simultaneous sockets that the server accepts. This is an advanced option and should not be used.

string_length

Length of payload to/from the NRPE agent. This is a hard specific value so you have to "configure" (read recompile) your NRPE agent to use the same value for it to work.

Default

1024

NRPE Handler Section

This is a wrapper page the actual data is on the following page [NRPEListener/config/nrpe_handlers](#)

1. 1. 1. 1. [Ovreview](#)
 1. [Alias \(builtin commands\)](#)
 2. [NRPE NT Syntax](#)

Ovreview

DEPRECATED This part of the module is deprecated and should not be used. Refer to the [\[CheckExternalScripts\]](#) module instead. This module can add two types of command handlers.

First there are external command handlers that execute a separate program or script and simply return the output and return status from that. The other possibility is to create an alias for an internal command.

To add an external command you add a command definition under the `?NRPE Handlers?` section. A command definition has the following syntax:

```
[NRPE Handlers]
command_name=/some/executable with some arguments
test_batch_file=c:\test.bat foo $ARG1$ bar
command[check_svc]=inject CheckService checkAll
```

The above example will on an incoming `?test_batch_file?` execute the `c:\test.bat` file and return the output as text and the return code as the Nagios status.

Alias (builtin commands)

To add an internal command or alias is perhaps a better word. You add a command definition under the `?NRPE Handlers?` section. A command definition with the following syntax:

```
command_name=inject some_other_command with some arguments
check_cpu=inject checkCPU warn=80 crit=90 5 10 15
```

The above example will on an incoming `?check_cpu?` execute the internal command `?checkCPU?` with predefined arguments give in the command definition.

NRPE_NT Syntax

To leverage existing infrastructure you can copy your old definitions from NRPE_NT as-is. Thus the following:

```
command[check_svc]=inject CheckService checkAll
```

translates into a command called `check_svc` with the following definition:

```
CheckService checkAll
```