Configuration

Configuration is fairly simple and straight forward. Open the configuration file in notepad (or your favorite editor) "notepad <installation path>\NSC.ini" and edit it accordingly. A longer description of the Configuration file is included in the following page.

The file has sections (denoted with section name in brackets) and key/value pairs (denoted by key=value). Thus it has the same syntax as pretty much any other INI file in windows.

The sections are described in short below. The default configuration file has a lot of examples and comments so make sure you change this before you use NSClient++ as some of the examples might be potential security issues.

The configuration can also be stored in the system registry (HKLM\Software\NSClient++) there is currently no UI to configure this so the simplest way is to maintain the configuration in the INI file and "Migrate that" to the registry. This is can be done via the [RemoteConfiguration] module but in short:

```
NSClient++ -noboot RemoteConfiguration ini2reg
```

A sample configuration file is included in the download but can also be found here trunk/NSC.dist

Modules

This is a list of modules to load at startup. All the modules included in this list has to be NSClient++ modules and located in the modules subdirectory. This is in effect the list of plug-ins that will be available as the service is running. For information on the various plug-ins check the Modules section in the navigation box.

A good idea here is to disable all modules you don?t actually use for two reasons. One less code equals less potential security holes and two less modules means less resource drain.

A complete list of all available modules:

- CheckDisk (module)
- CheckEventLog (module)
- CheckExternalScripts (module)
- CheckHelpers (module)
- CheckSystem (module)
- CheckTaskSched (module)
- CheckWMI (module)
- FileLogger (module)
- LUAScript (module)
- NRPEListener (module)
- NSCAAgent (module)
- NSClientListener (module)
- RemoteConfiguration (module)
- SysTray (module)

Settings

This section has generic options for how NSClient++will work, some of these settings (such as allowed_hosts) is inherited in sections below so it is probably a better idea to set them here in the "global" section.

The options you have available here are

Option	Default value	Description
obfuscated_password	...	An obfuscated version of password. For more details refer to the password option below. To create the obfuscated Password use: "NSClient++.exe /encrypt"
password	...	The password used by various (presently only NSClient) daemons. If no password is set everyone will be able to use this service remotely.
allowed_hosts	127.0.0.1	A list (comma separated) with hosts that are allowed to connect and query data. If this is empty all hosts will be allowed to query data. BEWARE: NSClient++ will not resolve the IP address of DNS entries if the service is set to startup automatically. Use an IP address instead.
use_file	0	Has to be set to 1 if you want the file to be read (if set to 0, and the use_reg is set to 1 the registry will be used instead)

Advanced options:

Option	Default value	Description
master_key	...	The secret "key" used when (de)obfuscating passwords.
cache_allowed_hosts	1	Used to cache looked up hosts if you check dynamic/changing hosts set this to 0.

includes

A list of other configuration files to include when reading this file. Might be useful if you have a very complex setup or want to have setting split up in segments.

Module Configuration

NRPE Listener Sections

NRPE Section

This is section is included from 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 Handlers Section

This is section is included from 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:

```
CheckServcice checkAll
```

File Logging Sections

Log Section

This is section is included from the following page FileLogger/config

1. 1. 1. 1. Overview
    1. debug
    2. file
    3. date_mask
    4. root_folder

Overview

This section has options for how logging is performed with the [FileLogger] module. First off notice that for logging to make sense you need to enable the ?FileLogger.dll? module that logs all log data to a text file in the same directory as the NSClient++ binary if you don?t enable any logging module nothing will be logged.

The options you have available here are

Option	Default	Description
debug	0	A Boolean value that toggles if debug information should be logged or not. This can be either 1 or 0.
file	nsclient.log	The file to write log data to. If no directory is used this is relative to the NSClient++ binary.
date_mask	%Y-%m-%d %H:%M:%S	The date format used when logging to a file
root_folder	exe	Root folder if not absolute

debug

A Boolean value that toggles if debug information should be logged or not. This can be either 1 or 0.

Default

0

Values

Value	Meaning
0	Don't log debug messages

1	Log debug messages

file

The file to write log data to. If no directory is used this is relative to the NSClient++ binary.

Default
nsclient.log

date_mask

The date format used when logging to a file

Default
%Y-%m-%d %H:%M:%S

root_folder

Root folder if not absolute

Default
exe
Values

local-app-data	The file system directory that contains application data for all users. A typical path is C:\Documents and Settings\All Users\Application Data. This folder is used for application data that is not user specific. For example, an application can store a spell-check dictionary, a database of clip art, or a log file in the CSIDL_COMMON_APPDATA folder. This information will not roam and is available to anyone using the computer.
exe	Location of NSClient++ binary

NSClient Sections

NSClient Section

This is section is included from the following page NSClientListener/config

1. 1. 1. 1. Ovreview
    1. port
    2. obfuscated_password
    3. password
    4. allowed_hosts
    5. bind_to_address
    6. socket_timeout
    7. socket_back_log
    8. version

Ovreview

This is the [NSClientListener] module configuration options.

Option		Description

	Default value	
port	12489	The port to listen to
obfuscated_password		An obfuscated version of password.
password		The password that incoming client needs to authorize themselves by.
allowed_hosts		A list (coma separated) with hosts that are allowed to connect to NSClient++ via NSClient protocol.
socket_timeout	30	The timeout when reading packets on incoming sockets.

Advanced options:

Option	Default value	Description
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.
bind_to_address		The address to bind to when listening to sockets, useful if you have more then one NIC/IP address and want the agent to answer on a specific one.
version	auto	The version number to return for the CLIENTVERSION check (useful to "simulate" an old/different version of the client, auto will be generated from the compiled version string inside NSClient++

port

The port to listen to

Default
12489

obfuscated_password

An obfuscated version of password. For more details refer to the password option below.

Default
Empty string whjich means we will use the value from password instead.

password

The password that incoming client needs to authorize themselves by. This option will replace the one found under Settings for NSClient. If this is blank the option found under Settings will be used. If both are blank everyone will be granted access.

Default
Empty string whjich means we will use the value from password in the [Settings] section instead.

allowed_hosts

A list (coma separated) with hosts that are allowed to poll information from NSClient++. This will replace the one found under Setting for NSClient 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.

BEWARE: NSClient++ will not resolve the IP address of DNS entries if the service is set to startup automatically. Use an IP address instead or set cache_allowed_hosts=0 see above.

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

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)

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

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.

version

The version number to return for the CLIENTVERSION check (useful to "simulate" an old/different version of the client, auto will be generated from the compiled version string inside NSClient++

Values:

If given any str4ing will be returned unless auto in which case the proper version will be returned

Default
auto

Check System Sections

CheckSystem Section

This is section is included from the following page CheckSystem/config

1. 1. 1. 1. Overview
    1. CPUBufferSize
    2. CheckResolution?

3. auto_detect_pdh
4. dont_use_pdh_index
5. force_language
6. ProcessEnumerationMethod?
7. check_all_services[<key>]
8. MemoryCommitLimit?
9. MemoryCommitByte?
10. SystemSystemUpTime?
11. SystemTotalProcessorTime?
12. debug_skip_data_collection

Overview

The configuration for the CheckSystsem? module should in most cases be automagically detected on most versions of windows (if you have a problem with this let me know so I can update it). Thus you no longer need to configure the advanced options. There is also some other tweaks that can be configured such as check resolution and buffer size.

Option	Default value	Description
CPUBufferSize	1h	The time to store CPU load data.
CheckResolution?	10	Time between checks in 1/10 of seconds.

Advanced options:

Option	Default value	Description
auto_detect_pdh	1	Set this to 0 to disable auto detect (counters.defs) PDH language and OS version.
dont_use_pdh_index	0	Set this to 1 if you dont want to use indexes for finding PDH counters.
force_language		Set this to a locale ID if you want to force auto-detection of counters from that locale.
ProcessEnumerationMethod?	auto	Set the method to use when enumerating processes PSAPI, TOOLHELP or auto
check_all_services[<key>]	ignored	Set how to handle services set to <key> state when checking all services
MemoryCommitLimit?	\Memory\Commit Limit	Counter to use to check upper memory limit.
MemoryCommitByte?	\Memory\Committed Bytes	Counter to use to check current memory usage.
SystemSystemUpTime?	\System\System Up Time	Counter to use to check the uptime of the system.
SystemTotalProcessorTime?	\Processor(_total)\% Processor Time	Counter to use for CPU load.
debug_skip_data_collection	0	**DEBUG** Used to disable collection of data

CPUBufferSize

The time to store CPU load data. The larger the buffer the more memory is used. This is a time value which takes an optional suffix for which time denominator to use:

Suffix	Meaning
s	second

m	minutes
h	hour
d	day

Default
1h

CheckResolution?

Time between checks in 1/10 of seconds.

Default
10

auto_detect_pdh

Set this to 0 to disable auto detect (counters.defs) PDH language and OS version.

Values

Value	Meaning
0	Don't attempt automagically detect the counter names used.
1	Use various menthods to figure out which counters to use.

Default
1

dont_use_pdh_index

When autodetecting counter names do **NOT** use index to figure out the values.

Values

Value	Meaning
0	Use indexes to automagically detect the counter names used.
1	Do **NOT** use indexes to figure out which counters to use.

Default
0

force_language

When index detection fails your local is used. Here you can override the default local to force another one if the detected local is incorrect.

Values
Any locale string like SE_sv (***not sure here haven't used in years***)
Deafult
Empty string which means the system local will be used.

ProcessEnumerationMethod?

DEPRECATED Set the method to use when enumerating processes PSAPI, TOOLHELP or auto No longer used (only PSAPI is supported).

check_all_services[<key>]

When using check all in a service check the default behaviour is that service set to auto-start should be started and services set to disabled should be stopped. This can be overridden using this option. Keys avalible:

Key	Default	Meaning
SERVICE_BOOT_START	ignored	TODO
SERVICE_SYSTEM_START	ignored	TODO
SERVICE_AUTO_START	started	TODO
SERVICE_DEMAND_START	ignored	TODO
SERVICE_DISABLED	stopped	TODO

MemoryCommitLimit?

Counter to use to check upper memory limit.

Default
\Memory\Commit Limit

MemoryCommitByte?

Counter to use to check current memory usage.

Default
\Memory\Committed Bytes

SystemSystemUpTime?

Counter to use to check the uptime of the system.

Default
\System\System Up Time

SystemTotalProcessorTime?

Counter to use for CPU load.

Default
\Processor(_total)\% Processor Time

debug_skip_data_collection

DEBUG Used to disable collection of data

Default
0

External Script Sections

External Script Section

This is a wrapper page the actual data is on the following page CheckExternalScripts/config/external_script

1. 1. 1. 1. Overview
    1. command_timeout
    2. allow_arguments
    3. allow_nasty_meta_chars
    4. script_dir

Overview

Configure how the External Scripts module works (not to be confused with the "External Scripts" section below that holds scripts that can be run.

Option	Default value	Description
command_timeout	60	The maximum time in seconds that a command can execute.
allow_arguments	0	A Boolean flag to determine if arguments are accepted on the command line.
allow_nasty_meta_chars	0	Allow NRPE execution to have ?nasty? meta characters that might affect execution of external commands.
script_dir		When set all files in this directory will be available as scripts. **WARNING**

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.

Values:

Any number (positive integer) representing time in seconds.

Default

60 (seconds).

Example

Set timeout to 120 seconds

```
[External Script]
command_timeout=120
```

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.

Values

Value	Meaning
0	Disallow arguments for commands
1	Allow arguments for commands

Default

0 (false).

Example

Allow arguments

```
[External Script]
allow_arguments=1
```

allow_nasty_meta_chars

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

Values

This list contain all possible values

Value	Meaning
0	Disallow nasty arguments for commands
1	Allow nasty arguments for commands

Default

0 (false)

Example

Allow nasty arguments

```
[External Script]
allow_nasty_meta_chars=1
```

script_dir

When set all files in this directory will be available as scripts. This is pretty dangerous but can be a bit useful if you use many scripts and you are sure no one else can add files there.

Value

Any directory (can be relative to NSClient++)

Default

Empty (meaning no scripts are added)

Example

All scripts ending with bat in the scripts folder (of NSClient++ installation directory) will be added as scripts.

```
[External Script]
script_dir=.\scripts\*.bat
```

External Scripts Section

This is a wrapper page the actual data is on the following page CheckExternalScripts/config/external_scripts

1. 1. 1. 1. Overview

Overview

A list of scripts and their aliases available to run from the CheckExternalScripts module. Syntax is: **<command>=<script> <arguments>** for instance:

```
check_es_long=scripts\long.bat
check_es_ok=scripts\ok.bat
check_es_nok=scripts\nok.bat
check_vbs_sample=cscript.exe //T:30 //NoLogo scripts\check_vb.vbs
check_es_args=scripts\args.bat static $ARG1$ foo
```

To configure scripts that request arguments, use the following syntax:

```
check_script_with_arguments=scripts\script_with_arguments.bat $ARG1$ $ARG2$ $ARG3$
```

Use ./check_nrpe ... -c check_script_with_arguments -a arg1 arg2 arg3 ... Make sure you type $ARG1$ and not $arg1$ (case sensitive)

NOTICE For the above to work you need to enable allow_arguments in **both** NRPEListener and CheckExternalScripts!

External Alias Section

This is a wrapper page the actual data is on the following page CheckExternalScripts/config/external_alias

1. 1. 1. 1. Overview

Overview

A simple and nifty way to define aliases in NSClient++. Aliases are good for defining commands locally or just to simply the nagios configuration. There is a series of "useful" aliases defined in the included configuration file which is a good place to start. An alias is an internal command that has been "wrapped" (to add arguments). If you want to create an alias for an external command you can do so but it still needs the normal defnition and the alias will use the internal alias of the external command.

WARNING Be careful so you don't create loops (ie check_loop=check_a, check_a=check_loop)

```
[External Aliases]
alias_cpu=checkCPU warn=80 crit=90 time=5m time=1m time=30s
alias_disk=CheckDriveSize MinWarn=10% MinCrit=5% CheckAll FilterType=FIXED
alias_service=checkServiceState CheckAll
alias_mem=checkMem MaxWarn=80% MaxCrit=90% ShowAll type=physical
```

Event Log Sections

Event Log Section

This is section is included from the following page CheckEventLog/config

EventLog?

The [EventLog?] section is used by the CheckEventLog module.

Advanced options:

Option	Default	Description
debug	0	Log all "hits" and "misses" on the eventlog filter chain, useful for debugging eventlog checks but very very very noisy so you don't want to accidentally set this on a real machine.
buffer_size	65536	Sets the buffer memory size used by NSClient++ when processing event log check commands. For details see below.

debug

Used to log all information regarding hits and misses on the filtering,. This has sever performance impact as well as log file will grow so do not use unless you are debugging.

```
[EventLog]
debug=1
```

buffer_size

This option was added in version 3.4

This parameter is set in the nsc.ini file and needs to be put under a heading of [EventLog?] (this heading may need to be created). The buffer reserves memory each time an eventlog check is being run when so set the size accordingly (or you will be wasting lots of memory).

To change the default setting of 64KB add (or edit) in the nsc.ini file an entry for buffer size (buffer_size=512000) where the value is in bytes. Often times the buffer size will need to be increased when using the %message% variable in return results. Most often you only need to increase this if you get error reported in the log file from NSClient++

```
[EventLog]
buffer_size=512000
```

Complete configuration

This are the default values for the entire EventLog? section

```
[EventLog]
debug=0
buffer_size=64000
```

NSCA Agent Sections

This page describes the configuration options for the NSCA module.

NSCA Agent Section

This is a wrapper page the actual data is on the following page NSCAAgent/config/NSCA_Agent

1. 1. 1. 1. Ovreview
    1. interval
    2. nsca_host
    3. nsca_port
    4. encryption_method
    5. password
    6. hostname
    7. debug_threads

Ovreview

Options to configure the NSCA module.

Option	Default value	Description
interval	60	Time in seconds between each report back to the server (cant as of yet be set individually so this is for all "checks")
nsca_host	...	The NSCA/Nagios(?) server to report results to.
nsca_port	5667	The NSCA server port
encryption_method	1	Number corresponding to the various encryption algorithms (see below). Has to be the same as the server or it wont work at all.
password		The password to use. Again has to be the same as the server or it won't work at all.

Advanced options:

Option	Default value	Description
hostname		The host name of this host if set to blank (default) the windows name of the computer will be used.
debug_threads	1	**DEBUG**Number of threads to run, no reason to change this really (unless you want to stress test something)

interval

Time in seconds between each report back to the server (cant as of yet be set individually so this is for all "checks")

Value

Any positive integer (time in seconds)

Default

60 (seconds)

nsca_host

The NSCA/Nagios(?) server to report results to.

Values
: Hostname or IP address to submit back results to.

Default
: Empty string (will in 3.7 and above mean don't submit results)

nsca_port

The NSCA server port

Values
: Any positive integer (port number ought to be less then 65534)

Default
: 5667

encryption_method

Number corresponding to the various encryption algorithms (see below). Has to be the same as the server or it wont work at all.

Values

Supported encryption methods:

#	Algorithm
0	None (**Do NOT use** this option)
1	Simple XOR (**No security, just obfuscation**, but very fast)
2	DES
3	3DES (Triple DES)
4	CAST-128
6	xTEA
8	BLOWFISH
9	TWOFISH
11	RC2
14	RIJNDAEL-128 (AES)
20	SERPENT

Default
: 1 (I am note sure I thought default was 14?)

password

The password to use. Again has to be the same as the server or it won't work at all.

Values
: Any string (should be the same as the one configured in nsca.conf

hostname

The host name of this host if set to blank (default) the windows name of the computer will be used.

Values
Any string (or auto)
Default
auto (means windows hostname will be used)

debug_threads

DEBUGNumber of threads to run, no reason to change this really (unless you want to stress test something)

Values
Any positive integer larger then or equal to 1
Default
1

NSCA Commands Section

This is a wrapper page the actual data is on the following page NSCAAgent/config/NSCA_Commands

1. 1. 1. 1. Overview

Overview

A list of commands to run and submit each time we report back to the NSCA server. A command starting with host_ will be submitted as a host command. For an example see below: This will report back one service check (called my_cpu_check) and one host check (host checks have no service name).

```
[NSCA Commands]
my_cpu_check=checkCPU warn=80 crit=90 time=20m time=10s time=4
host_check=check_ok
```

LUA Scripts

A list of LUA script to load at startup. In difference to "external checks" all LUA scripts are loaded at startup. Names have no meaning since the script (on boot) submit which commands are available and tie that to various functions.

```
[LUA Scripts]
scripts\test.lua
```