Configuring And Using The check_ncpa Beta Plugin

Purpose

This document describes how to set up the Version 1.2.4 - BETA implementation of the check_ncpa check plugin written in C for Nagios XI and Nagios Core. This plugin is used to run checks on hosts running the Nagiso Cross Platform (NCPA) agent.

Description

check_ncpa is a feature-matching C replacement for the Python plugin check_ncpa.py that ships with Nagios XI. It communicates with the NCPA REST API over HTTPS and returns standard Nagios exit codes (OK, WARNING, CRITICAL, UNKNOWN).

This plugin accepts the same command-line arguments, produces the same output format, and returns the same exit codes as check_ncpa.py, making it a drop-in replacement. The C implementation eliminates the Python runtime dependency, reduces execution overhead, and produces a single statically-linkable binary.

Supported features (matching check_ncpa.py):

- Metric checks with warning/critical thresholds (`-w`, `-c`)
- Unit prefix conversion (`-u`) and unit override (`-n`)
- Custom plugin execution with arguments (`-a`)
- Delta/rate checks (`-d`)
- Metric listing mode (`-l`)
- Token-based authentication (`-t`)
- Configurable port (`-P`) and timeout (`-T`)
- SSL certificate verification (`--secure`)
- Performance data injection (`--performance`)
- Extra query arguments (`--queryargs`)
- Verbose and debug output (`-v, -D`)
- NCPA < 2 response format compatibility (stdout/returncode swap handling)

Installing Dependencies

Two development libraries are required to build the plugin:

- **libcurl** - HTTP client library
- **libcjson** - Lightweight JSON parser

Run the following commands to install the dependencies:

RHEL / CentOS / Oracle:

```
sudo yum install gcc make libcurl-devel cjson-devel pkg-config
```

Or with dnf:

```
sudo dnf install gcc make libcurl-devel cjson-devel pkg-config
```

Debian / Ubuntu:

```
sudo apt-get install gcc make libcurl4-openssl-dev libcjson-dev pkg-config
```

Building the Plugin

From the directory containing the source files, run the following commands:

```
make clean
make
```

This produces the binary `check_ncpa`.

To verify the build, run these commands:

```
./check_ncpa --version
./check_ncpa --help
```

Installing the Compiled Binary

Install the compiled binary into the Nagios plugin directory with the following command:

```
sudo make install
```

By default this installs to `/usr/local/nagios/libexec/.` To install to a different directory (for example, the 64-bit plugin path on RHEL-based systems):

```
sudo make install INSTALL_DIR=/usr/lib64/nagios/plugins
```

Verify the installed binary by running:

```
/usr/local/nagios/libexec/check_ncpa --version
```

It is recommended to keep the original Python plugin as a backup with this command:

```
sudo cp /usr/local/nagios/libexec/check_ncpa.py /usr/local/nagios/libexec/check_ncpa.py.bak
```

Configuring Nagios XI to use the C Command

The `check_xi_ncpa` command in Nagios XI calls `check_ncpa.py` by default. To switch it to use the compiled C binary, update the command definition using one of the methods below.

The default command definition is:

```
define command {
    command_name    check_xi_ncpa
    command_line    $USER1$/check_ncpa.py -H $HOSTADDRESS$ $ARG1$
}
```

This needs to be changed to:

```
define command {
    command_name    check_xi_ncpa
    command_line    $USER1$/check_ncpa -H $HOSTADDRESS$ $ARG1$
}
```

The only change is replacing `check_ncpa.py` with `check_ncpa`.

Changing the Check Command

The check command can be modified via the Nagios XI web UI.

To do so, execute the following steps:

1. Log into Nagios XI as an administrator from your web browser.
2. Navigate to: **Configure > Core Config Manager (CCM)**
3. 3. In the left-hand menu, click **Commands**.
4. 4. Search for **check_xi_ncpa** and click on it to edit.
5. 5. In the "Command Line" field, change:

```
$USER1$/check_ncpa.py -H $HOSTADDRESS$ $ARG1$
```

...to:

```
$USER1$/check_ncpa -H $HOSTADDRESS$ $ARG1$
```

6. Click **Save**.
7. Click **Apply Configuration** in the banner that appears at the top of the page.

This validates and activates the change.

Verifying Your Changes

After making the change, you'll want to verify that checks are working correctly.

Option 1 - Command Line

Run a manual check from the command line of your Nagios XI server:

```
/usr/local/nagios/libexec/check_ncpa -H <target_host> -t <token> -M cpu/percent -w 80 -c 90
```

You should see output like:

```
OK: Percent was 12.5%|'percent'=12.5%;80;90
```

Option 2 - Web Interface

In the Nagios XI web interface, navigate to the Status Detail page of a host or service that uses the check_xi_ncpa command. Click **Force an immediate** check to force an immediate check and confirm results are returned correctly.

Rolling Back the Plugin

In the event that you wish to revert to the Python plugin, you can do so via the Nagios XI web interface. Log in as an administrator, then edit the check_xi_ncpa command in the CCM and change the command line back to:

```
$USER1$/check_ncpa.py -H $HOSTADDRESS$ $ARG1$
```

Click **Save**, then **Apply Configuration**.

Usage Examples

You can run the following example commands from the `/usr/local/nagios/libexec/` directory.

Check CPU usage

```
./check_ncpa -H 192.168.1.100 -t mytoken -M cpu/percent -w 20 -c 40
```

Check memory with unit prefix

```
./check_ncpa -H 192.168.1.100 -t mytoken -M memory/virtual -w 80 -c 90 -u G
```

Run a custom plugin on the remote host

```
./check_ncpa -H 192.168.1.100 -t mytoken -M plugins/check_load.sh -a '-w 3 -c 5'
```

List all available metrics on a host

```
./check_ncpa -H 192.168.1.100 -t mytoken -l
```

Check disk usage on a specific mount

```
./check_ncpa -H 192.168.1.100 -t mytoken -M 'disk/logical/C:|' -w 80 -c 90
```

Check with SSL certificate verification enabled

```
./check_ncpa -H 192.168.1.100 -t mytoken -M cpu/percent -w 80 -c 90 -s
```

Key Nagios XI File Paths

Plugin directory ($USER1$): /usr/local/nagios/libexec/

Command definitions (CCM): /usr/local/nagios/etc/commands.cfg

Static config directory: /usr/local/nagios/etc/static/

Resource file ($USER1$ macro): /usr/local/nagios/etc/resource.cfg

Main Nagios config: /usr/local/nagios/etc/nagios.cfg