NCAP
“Nagios Collector and Plugin”

version 0.4

Giray Devlet <giray@osc.nl>

2004-02-29

Table of Contents

Introduction

This document is about using and configuring the Nagios Collector and Proxy (ncap). You need an understanding of

- Nagios - http://www.nagios.org/
- nrpe (Nagios Remote Plugin Executor) - http://www.nagios.org/download/extras.php

ncap will receive requests by check_ncap, via the nagios (monitoring) host. Then these requests are proxied to actual nrpe daemons.

Design

The idea is to have a separate collector that gathers information from nrpe's. Then nagios gets its information from the collector (via a separate plugin, similar to check_nrpe). The need for something like this comes from the need for scalability (to have separate collectors for separate environments). Or to use it as a gateway between network borders.

'*check_ncap*' si a plugin for nagios, it:

communicates with a given collector. Has parameters such as collector-IP, host-IP, check-command.

'*ncap*' si a daemon that can work in two modes

I) transparent

In transparent mode it will receive requests from a specified nagios host, and blindly forward the request to the specified host with the given command.

Received requests will be cached, and continuously (according to a specified interval) checked. Results will be cached at the collector. When nagios asks for a status, cached information will be provided.

If for a certain period new requests do not arrive, the checks for that service will not be repeated.

Advantage:

- No need for extra configuration on the collector. (only specification of authorized nagios host)

Disadvantage:

- nagios host IP's can be spoofed and random checks can be executed, which can result into DOS attacks!
- initial check will result in a 'UNKNOWN' state.

II) non-transparent

In the Non-transparent mode extra configuration is required. Host / command pairs need to be entered to specify which checks are allowed.

Checks will continue independent from requests generated by the nagios host. When a request is made, cached information will be provided.

Advantage:

- In case of a DOS attack, monitored systems will not be directly bothered.

Disadvantage:

•One more system will have to be configured.

Internals

ncap consists out of 3 threads.

1) The listener thread

2) The scheduler thread

3) The worker thread

When the program is initiated the ncap.cfg file is read in and processed. Then the listener and scheduler threads are stared.

The listener thread will listen to incoming *check_ncap* requests at a given port. The default ncap port is 5667. When a request comes in, an new thread (worker) is created which handles the connection, and returns information to *check_ncap*. If the request is a previously unknown request, it is added to the *command_list*, and *check_ncap* gets an 'UNKNOWN' state with a message saying that the 'command'/' request' has been scheduled. If it is a previously known request, then the latest information is retrieved from the *command_list* and sent back to *check_ncap*.

The scheduler thread traverses the command_list and sends requests out to nrpe daemons, masquerading as a check_nrpe request. The results are stored in the command_list. If a command has not been checked for a certain period of time (*set by old_age in the configuration file*) it is removed from the command_list.

At the end of each complete traversal the scheduler looks at how long it took to get trough the whole list. If the time is less then the *scheduler_wait* value then it waits before processing anything else. When the the *scheduler_wait* time has been reached or passed, the scheduler starts over.

Installation

Installation can be made from source or via a pre-compiled RPM. Both of which can be found at http://thelinuxplatform.nl/ncap

Installation from Source

After you have downloaded the source :

```
# tar zxvf ncap-0.4.tar.gz
# cd ncap-0.4
# ./configure
checking for a BSD-compatible install... /usr/bin/install -c
checking for gcc... gcc
checking for C compiler default output... a.out
checking whether the C compiler works... yes
checking whether we are cross compiling... no

[ ... output cut ... ]

config.status: creating Makefile
config.status: creating src/Makefile
config.status: creating subst
config.status: creating src/config.h


*** Configuration summary for ncap 0.2 10-02-2004 ***:

 General Options:
 -------------------------
 NCAP port:  5667
 NRPE port:  5666
 NCAP user:  nagios
 NCAP group: nagios
 NCAP daemon installation: /usr/local
 NCAP client installation: /usr/local


Review the options above for accuracy.  If they look okay,
type 'make all' to compile the NCAP daemon and client.

#
```

Currently 'configure' will happily announce that everything went ok. Please check if that is really so.

Now we can type 'make all'

```
# make all
cd ./src/; make ; cd ..
make[1]: Entering directory `/home/giray/OLD/ncap/gd/ncap-0.4/src'
gcc  -g  -O2  -I/usr/include/openssl  -I/usr/include  -DHAVE_CONFIG_H  -o  ncap  ncap.c
utils.c ssl_thread_safe.c -L/usr/lib  -lpthread -lssl -lcrypto -lnsl
gcc  -g  -O2  -I/usr/include/openssl  -I/usr/include  -DHAVE_CONFIG_H  -o  check_ncap
check_ncap.c utils.c -L/usr/lib  -lpthread -lssl -lcrypto -lnsl
make[1]: Leaving directory `/home/giray/OLD/ncap/gd/ncap-0.4/src'

*** Compile finished ***

If the NCAP daemon and client compiled without any errors, you
can continue with installation.  The NCAP daemon and client
binaries are located in the src/ subdirectory.

 ** If this is your monitoring host **

     - Copy the check_ncap client to the directory that
       contains your Nagios plugins.
     - Create a command definition in your Nagios config
       file for the NCAP client.  See the README file for
       more info on doing this.
 ** If this host will be running the NCAP daemon **

     - Copy the ncap daemon to /usr/sbin, /usr/local/nagios
       or wherever you feel it fits best.
     - Copy the sample ncap.cfg config file to /etc,
       /usr/local/nagios or wherever you feel it fits best.
```

At this point you can copy

```
./src/ncap
./src/check_ncap
ncap.cfg
```

to relevant locations like

```
cp ./src/ncap /usr/sbin
cp ./src/check_ncap /usr/lib/nagios/plugins
cp ncap.cfg /etc/nagios
```

alternatively you can use '*make daemon_install*' to install ncap, or use '*make client_install*' to install check_ncap.

To uninstall you can also use '*make uninstall*', however, this will remove any ncap related files on your system!!!

Installation with RPM

There are two ncap RPMs. The first one installs the ncap daemon which is supposed to run on the Collector/Proxy and is called:

ncap-x.y.z-n.i386.rpm

The second one is to be used on the nagios host and is called:

ncap-plugin-x.y.z-n.i386.rpm

NCAP machine

So on the ncap machine the following has to be done

```
rpm -Uvh ncap-0.4-1.i386.rpm
```

Then you have to edit the *ncap.cfg* file which should be at */etc/nagios/ncap.cfg*. More information about this in the next section

NAGIOS machine

The following has to be done to install check_ncap

```
rpm -Uvh ncap-plugin-0.4-1.i386.rpm
```

This will install *check_ncap* in */usr/lib/nagios/plugins*

However, for nagios to be able to use some files in the nagios configuration directory have to be edited. Details in the next section.

Configuring NCAP

After ncap and check_ncap have been installed some configuration has to be made to be able to use them.

The ncap daemon

The configuration file ncap is looking for should be either under */etc/nagios/ncap.cfg* or */usr/local/nagios/etc/ncap.cfg*. Of course since ncap is started with '*-c*' to tell it where its configuration file is, it could be virtually any place.

The configuration parameters are as follows:

server_port

This determines on which port ncap is listening to incoming requests. The default value is *5667*.

client_port

This determines which port the NRPE daemons are listening at, to which nrpe is going to make a connection. The default value is *5666*. It is currently not possible to connect to different port per NPRE instance. It is assumed that all nrpe's in one environment listen to the same port.

server_address

This tells ncap which address to bind to if there are multiple IP addresses on the same host. This is commented out by default.

allowed_hosts

This is a list of hosts that are allowed to connect to this ncap daemon. The default value is '*127.0.0.1*', the localhost. This is a comma delimited list. Please make sure that there are no white spaces between the commas.

ncap_user

This is the user name that ncap is going to run as. The default value is *nagios*.

ncap_group

This the group that ncap is going to run as. The default value is *nagios*.

log_level

The log level determines how much logging ncap will do. For a productional environment this should be 0. The log levels are as follows:

- 0 Errors
- 1 Informative messages
- 2 Some verbosity, useless information
- 4 Real debug stuff
- 8 Developer stuff, very verbose
- 16 Logs that even the developers hardly want

The Default value is *0*.

transparent_proxy

This determines whether ncap will work as transparent proxy, or in restrictive mode. While working as transparent proxy all requests from valid hosts are accepted. In restrictive mode only commands defined in ncap.cfg will be accepted. (*restrictive mode has not been implemented yet in 0.4.x*)

old_age

This determines when a command is considered to *old* and is removed from the command_list. Each command in the command_list has a timestamp, which is updated every time there is a request for it. If a command is not asked for, for more then the time defined by old_age it is dropped. Time is given in seconds and the default value is *600* (10 minutes).

scheduler_wait

This is the maximum amount of time the scheduler will sleep before it starts to go trough the command list again. If processing the list takes more then the value of scheduler_wait, then it will not wait but start another round immediately. In case the queue is processed faster, then it will wait the difference between the scheduler_wait and processing time values. Time is given in seconds and the default value is *60*. (1 minute).

Since the scheduler currently is single threaded, in case it cannot reach a nrpe host it will wait for a 10 second timeout per command/request. This will slow down the process considerably.

include

This can be used to include an external configuration file with the same format as ncap.cfg. It is commented out on default. (*not tested*).

include_dir

This can be used to include a configuration directory, which contains configuration files (*not tested*).

command

This is used for either the restrictive proxy mode, or to quicken the processing of commands at start up.

Syntax:

command[check_disk1][192.168.1.1]

The first field has the command name that the nrpe daemon will respond to, the second field is the IP address of the nrpe daemon. This is commented out on default. **This functionality has not been implemented yet, and is scheduled for the 0.6 release.**

The check_ncap plugin

The check_ncap plugin is used to send requests from a nagios host to the ncap daemon. For nagios to be able to use check_ncap several things have to be done.

First the following has to be added to the checkcommands.cfg file:

```
define command{
 command_name    check_ncap
  command_line    $USER1$/check_nrcap -H $HOSTADDRESS$ -C $ARG1-c $ARG2$
}
```

Then you can use this from within services.cfg like

```
define service{
  use                      generic-service
  host_name                myhost.foo.org
  service_description      disk1
  is_volatile              0
  check_period             24x7
  max_check_attempts       3
  normal_check_interval    5
  retry_check_interval     1
  contact_groups           disk-admins
  notification_interval    120
  notification_period      workhours
  notification_options     c,r
  check_command            check_ncap!my_collector!disk1
}
```

For a complex environment with many collectors an option would be to define multiple check commands as follows:

```
define command{
 command_name    check_ncap_env1
  command_line    $USER1$/check_nrcap -H $HOSTADDRESS$ -C env1_collector -c $ARG2$
}
define command{
 command_name    check_ncap_env2
  command_line    $USER1$/check_nrcap -H $HOSTADDRESS$ -C env2_collector -c $ARG2$
}
define command{
 command_name    check_ncap_env3
  command_line    $USER1$/check_nrcap -H $HOSTADDRESS$ -C env3_collector -c $ARG2$
}
```

Then in services we would have something like this

```
define service{
  use                        generic-service
  host_name                  myhost.foo.org
  service_description        disk1
  is_volatile                0
  check_period               24x7
  max_check_attempts         3
  normal_check_interval      5
  retry_check_interval       1
  contact_groups             disk-admins
  notification_interval      120
  notification_period        workhours
  notification_options       c,r
  check_command              check_ncap_env1!disk1
}
```

Debugging

Currently all log information goes to *local2*. You most likely have this not configured and will not get any log messages no matter how high you set the log_level.

To see the log messages you will have to edit */etc/syslog.conf*

```
local2.*=                                               /var/log/ncap.log
```

then you have to restart syslog.

For developers it is possible to run ncap without having daemonizing.

```
./ncap -D -c /etc/nagios/ncap.cfg
```

will leave ncap running in you current terminal.

Contact Info

To contact the developers you can write to

ncap-devel@yahoogroups.com

For up-to-date information please visit the website at

http://thelinuxplatform.nl/ncap/