

Debugging Nagios Plugins

Jess Portnoy

Kaltura, Inc

jess.portnoy@kaltura.com

September 30, 2015

Abstract

This session will discuss how to debug malfunctioning plugins, show real life situations in which the plugin does not behave as expected and ways to troubleshoot and resolve the issue.

We will also cover some basic security best practices and the potential issues that may arise when creating a secured setup.

Common plugin failure reasons

There are many reasons why a plugin would malfunction but here are some very common ones:

- Wrong file system permissions
- Networking issues
- Missing dependencies
- A massive amount of additional weird problems

Helpful tools

The following debugging tools will be covered in this presentation:

- Capture plugin
- strace
- telnet
- nmap
- netcat
- curl
- ldd

Debugging time

In this demo, we will debug malfunctioning plugins and fix things so that they work correctly:)

The following scenarios will be inspected:

- Nagios web interface fails to load
- Nagios does not send mail alerts
- `check_mysql` fails
- host check fails
- `check_http` returns the wrong SSL certificate info

When S*** happens...



Nagios web interface fails to load

Upon requesting the Nagios web interface, one gets 'Internal Server Error' [HTTP 500]

- Find the Nagios Apache configuration
- Check Apache error log for errors and hopefully, correct them:)

Trying to reschedule test execution fails

Upon committing, one gets: Error: Could not stat() command file
'/var/lib/nagios3/rw/nagios.cmd'!

- Check which user and group Apache is running as
- Add the Apache user to the nagios group so that Apache can write to it

No mail alerts are received

- Make sure notifications are enabled for the service
- Check the Nagios log to see which command is used to send mail alerts
- Make sure an MTA is running
- Try running the command manually from the shell as the nagios user and check the MTA log for errors

Host check fails cause ICMP is blocked

In many cases, this is out of your control, in such cases, the easiest thing to do is to set an alternative command for checking that the host is alive.

Use the `check_command` directive in your host/hostgroup definition, specifying an alternative command.

Capture command output

The `capture_output.pl` script is used as a wrapper that runs the actual command, stores the `STDOUT` and `STDERR` outputs to a log file and then passes the output and RC to Nagios.

If the original command's return code is bigger than 3 [UNKNOWN], 3 will be returned and the original return code will appear as part of the output.

'(Return code of 127 is out of bounds - plugin may be missing)'

Nagios can only handle 0,1,2,3 as return codes, anything else will result in the output above.

To debug this, lets use the `capture_output.pl` introduced in a previous slide.

Consider this command:

```
define command{  
  command_name check_ssl_cert_bad  
  command_line /usr/lib/nagios/plugins/check_special_http -H '$HOSTADDRESS$' -I  
    '$HOSTADDRESS$' -C10  
}
```

'(Return code of 127 is out of bounds - plugin may be missing)' - cont'd

We will now revise the `command_line` to use the capture wrapper, like so:

```
define command{  
  command_name check_ssl_cert_bad  
  command_line /usr/lib/nagios/plugins/capture_plugin.pl  
               /usr/lib/nagios/plugins/check_special_http -H '$HOSTADDRESS$' -I  
               '$HOSTADDRESS$' -C10  
}
```

'(Return code of 127 is out of bounds - plugin may be missing)' - cont'd

This will help us in two ways:

Nagios will now show the following output instead of '(Return code of 127 is out of bounds - plugin may be missing)':

Original RC: 127, /usr/lib/nagios/plugins/check_special_http: error while loading shared libraries: libssl.so.0.9.8: cannot open shared object file: No such file or directory

The captured-plugins.log will have an entry with the full command so we can try to run it in the shell and debug.

check_mysql plugin fails with Can't connect to MySQL server on 'mysql.host' (111)

- Use capture_output.pl to log the command output to a file
- Try running the command from the shell
- On the MySQL server, check what IP/network the daemon is binded with, and what port is the listener on
- Check mysql.user table to make sure the username and host Nagios uses is allowed



SSL Certificate check shows wrong certificate

Check returns with RC 0 (OK) and shows the certificate is has plenty time till it expires.

However, when looking at the certificate from a browser or using curl, a different certificate is displayed.

- Figure out how the check_http plugin performs its check. Since the plugin is a pre-compiled binary (written in C), strace would do nicely.
- Use the openssl CLI client to run a similar check from the shell

References

-  Nagios Capture Output Plugin - https://github.com/jessp01/debugging_nagios/blob/master/capture_output.pl
-  Scenarios and commands used in this session - https://github.com/jessp01/debugging_nagios
-  Nmap - <https://nmap.org>
-  cURL - <http://curl.haxx.se>
-  Nagios Exchange - <http://exchange.nagios.org>
-  Nagios Kaltura plugins - <http://exchange.nagios.org/directory/Utilities/Kaltura-monitors/details>
-  The home of the official Nagios Plugins - <https://nagios-plugins.org>

The End && questions