Nagios XI - Special Characters

Article Number: 580 | Rating: 3/5 from 2 votes | Last Updated: Thu, Aug 16, 2018 at 8:08 AM

Overview

Nagios XI, as well as the Linux command line interface, have some reserved characters that need to be escaped in specific ways.

Special characters are a group of characters that carry out special instructions, like the dollar sign `$` which does expansion in bash. If you tried to use a `$` in a password argument at will see it throw an unexpected error.

The term "escaped" or "escaping" is when you use a character like the backslash `\` to identifying that the next character should be treated as plain text, like the `$` in a password.

In Nagios Configurations an exclamation mark `!` is used as an argument separator. If you used this in a password then the password will be split and it will change the amount of arg to the monitoring engine.

This KB article demonstrates this behaviour and provides some solutions.

Command Line Example

This example shows you a password of `$ecret!` being used with the `check_wmi_plus.pl` plugin:

```
/usr/local/nagios/libexec/check_wmi_plus.pl -H 10.25.14.3 -u wmiagent -p $ecret! -m checkcpu -w '80' -c '90'
```

It produces this error message:

```
UNKNOWN - The WMI query had problems. You might have your username/password wrong or the user's access level is too low. Wmic error tex
[librpc/rpc/dcerpc_util.c:1290:dcerpc_pipe_auth_recv()] Failed to bind to uuid 4d9f4ab8-7d1c-11cf-861e-0020af6e7c57 - NT_STATUS_NET_WRI
[librpc/rpc/dcerpc_connect.c:790:dcerpc_pipe_connect_b_recv()] failed NT status (c0000022) in dcerpc_pipe_connect_b_recv
[wmi/wmic.c:196:main()] ERROR: Login to remote object.
NTSTATUS: NT_STATUS_ACCESS_DENIED - Access denied
```

Here is that example again but this time the `$` is being escaped:

```
/usr/local/nagios/libexec/check_wmi_plus.pl -H 10.25.14.3 -u wmiagent -p \$ecret! -m checkcpu -w '80' -c '90'
```

This time it worked:

```
OK (Sample Period 246 sec) - Average CPU Utilisation 0.34%|'Avg CPU Utilisation'=0.34%;80;90;
```

The next section uses this example in Core Configuration Manager.

Core Configuration Manager Example

This example shows you a password of `$ecret!` being used with the `check_wmi_plus.pl` plugin. Reading the previous section you know that you need to escape the `$` sign. This e the password is stored in the `$ARG2$` field for the service definition:

```
\$ecret!
```

However after making that change and applying configuration you still receive errors like:

```
UNKNOWN - The WMI query had problems. You might have your username/password wrong or the user's access level is too low. Wmic error tex
[librpc/rpc/dcerpc_util.c:1290:dcerpc_pipe_auth_recv()] Failed to bind to uuid 4d9f4ab8-7d1c-11cf-861e-0020af6e7c57 - NT_STATUS_NET_WRI
[librpc/rpc/dcerpc_connect.c:790:dcerpc_pipe_connect_b_recv()] failed NT status (c0000022) in dcerpc_pipe_connect_b_recv
[wmi/wmic.c:196:main()] ERROR: Login to remote object.
NTSTATUS: NT_STATUS_ACCESS_DENIED - Access denied
```

The reason for the error this time has to do with the `$` (again). Nagios is seeing the `$` and thinks this is the beginning of a macro / free variable, for example `$USER1$`. What you need with another `$` so Nagios knows it should be plain text, for example:

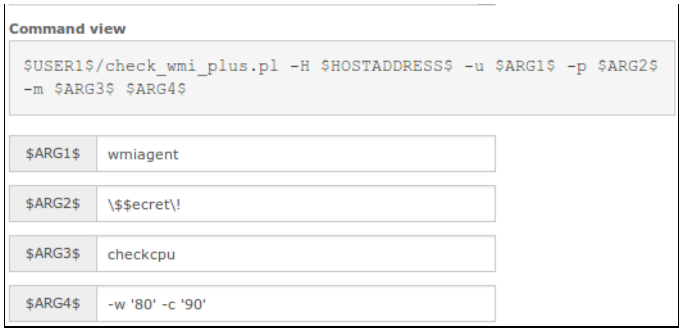
```
\$$ecret!
```

You still need the `\` as Nagios needs this for the command line. However after making that change and applying the configuration you may still get the same error:

The reason for the error this time has to do with the `!`. Nagios uses the `!` as an argument separator. What you need to do is escape the `!` with a `\` so Nagios knows it should be plair

```
\$$ecret\!
```

After applying the configuration the service now works. Nagios XI 5.4.0 and newer will automatically escape the `!` with a `\`. Here is a screenshot showing the final password stored in



As you can imagine special characters can make things complicated. The next section provides solutions for special characters.

Solutions

One solution is to define the password in a user macro in `resources.cfg` that can be used in your object definitions. This provides an additional benefit of storing the password in a hides it from plain sight. Please refer to the following documentation for detailed steps on how to implement this solution:

Documentation - Understanding The User Macros Component

Some plugins such as `check_wmi_plus.pl` provide the ability to use a dedicated file to store credentials in. This solution is specific to the plugin being used, not all plugins provide refer to the following documentation for detailed steps on how to implement an authentication file with `check_wmi_plus.pl`:

Documentation - Monitoring Windows Using WMI

Object Name & Macro Output

While not specifically related to this article, there are two directives in the `nagios.cfg` file that dictate what characters cannot be used for object names and macro output. Please r documentation for detailed information:

Illegal Object Name Characters

Illegal Macro Output Characters

More Information

The following link provides detailed information on special characters:

http://mywiki.wooledge.org/BashGuide/SpecialCharacters

Final Thoughts

For any support related questions please visit the Nagios Support Forums at:

http://support.nagios.com/forum/

Posted by: **tlea** - Tue, Apr 25, 2017 at 1:46 AM. This article has been viewed 6796 times.

Online URL: https://support.nagios.com/kb/article/nagios-xi-special-characters-580.html