

Nagios. Macros globales.

Nagios nos proporciona la posibilidad de usar macros globales explícitamente definidas dentro de Nagios. Digamos que vienen a ser como variables globales. Algunos usos interesantes son:

- No tener que repetir largas cadenas de path a los plugins p.e.
- Almacenar usuarios y contraseñas necesarios para acceder a chequear ciertos servicios.
- Almacenar nuestras comunidades snmp para usarlas luego en los chequeos.

Su uso es casi obligatorio en el caso de las contraseñas. ¿Por qué? Porque si usamos estas directamente en la definición de objetos, en determinadas vistas del GUI de Nagios Core podremos verlas tal cual. Si usamos las variables en dichas vistas veremos sólo las variables referenciadas.

Nagios define las variables a usar en el archivo de configuración **resource.cfg**. En una instalación de Nagios, si instalamos los ejemplos veremos que en dicho archivo figuran ya al menos variables definidas para el directorio de los plugins como veremos en breve con unos ejemplos.

Se pueden usar hasta 32 (\$USER1\$ – \$USER32\$) macros de este tipo (\$USERX\$). Desgraciadamente no nos podemos salir de la nomenclatura “\$USERX\$” ya que solo funcionarán las variables definidas de esta forma.

El uso de dichas variables de usuario es perfecto para poder usar ficheros de configuración de comandos fácilmente transportables a otras instalaciones y fácilmente modificables en caso de cambios. También, como ya mencionamos, de esta forma ocultamos información a los CGIs ya que desde estos se ve la definición de los comandos pero no “traslada” las variables.

Ejemplo de resources de una instalación de Nagios realizada con OMD. Fichero **/opt/omd/sites/foo/etc/nagios**

```
#####  
# OMD settings, please use them to make your config  
# portable, but dont change them  
$USER1$=/omd/sites/foo/lib/nagios/plugins  
$USER2$=/omd/sites/foo/local/lib/nagios/plugins  
$USER3$=foo  
$USER4$=/omd/sites/foo  
#####  
# set your own macros here:  
#$USER5$=...
```

Ejemplo de **resource.cfg** en una instalación por defecto de paquetes de Nagios realizada en [Debian](#). Fichero **/etc/nagios3/resource.cfg**

```
#####  
# Sets $USER1$ to be the path to the plugins
```

```
$USER1$=/usr/lib/nagios/plugins
# Sets $USER2$ to be the path to event handlers
#$USER2$=/usr/lib/nagios/plugins/eventhandlers
# Store some usernames and passwords (hidden from the CGIs)
#$USER3$=someuser
#$USER4$=somepassword
#####
```

De hecho en los archivos de configuración de chequeos de comandos “check-commands”, la definición de commands las usarán de la siguiente manera:

```
command_line $USER1$/check_http -H $HOSTADDRESS$
```

Un uso habitual será también como comentamos en el paso de usuarios y contraseñas en comandos. El siguiente ejemplo está extraído de un command para un plugin que chequea servidores vSphere ESXi y que veremos próximamente:

```
define command{
command_name check_vmware_api_dc_host_cpu_use
command_line $USER1$/check_vmware_api -D $ARG1$ -u $USER11$ -p $USER12$ -H $HOSTALIAS$ -l cpu -s
usage -w $ARG2$ -c $ARG3$
}
```

En general, se pueden usar dichas macros en todas las definiciones de objetos de Nagios referenciándolas en la forma: \$USERx\$.

En otro artículo veremos el uso de más tipos de macros que el entorno de Nagios define por defecto y que podemos usar en nuestros objetos. Además también veremos el uso de macros explícitas definidas en los propios objetos.