

Implementación de un manejador de evento (Event Handler) en Nagios para mantener una conexión PPTP en Windows

Por: Juan Felipe Muñoz Fernández
<http://www.juanfelipe.net>

Introducción

Nagios se ha convertido en una de las herramientas más poderosas de monitoreo de infraestructura informática y de telecomunicaciones debido a su robustez y flexibilidad. Una de las características que más disfruta el administrador de Nagios es su arquitectura modular; en Nagios, el núcleo de monitoreo está conformado por diferentes plugins que desempeñan la labor principal del sistema: monitorear, avisar de cualquier evento o cambio que suceda a los objetos que están siendo monitoreados y disparar los correspondientes manejadores de eventos que lidian con los cambios detectados. En este artículo se expondrá una situación en la que Nagios monitorea la disponibilidad de una conexión VPN, cuando esta conexión queda por fuera de línea, es decir, cuando se desconecta, Nagios dispara un manejador de evento que se encarga de reconectar remotamente el enlace VPN.

Escenario de monitoreo

En la figura 1 se ilustra el escenario de los componentes que están siendo monitoreados por Nagios.

Las características de este escenario de monitoreo son las siguientes.

- Nagios se está ejecutando sobre el servidor que lleva esta misma etiqueta en la figura.
- Nagios monitorea la disponibilidad del canal de acceso a Internet del enrutador *R* y la disponibilidad de la conexión VPN del servidor *S*.
- La conexión VPN la inicia el servidor *S*.
- El servidor *S* es un servidor Windows Server 2003.
- El enrutador *R* dispone de una dirección IP pública conocida por el administrador de Nagios.
- El enrutador *R* es administrado por el administrador de Nagios.
- La disponibilidad de la conexión VPN depende directamente de que existan un canal de Internet activo en el enrutador *R*.
- Se supone que el canal que conecta a Nagios a Internet está disponible siempre.

Lógica del monitoreo

Para que exista la conexión VPN desde el servidor *S* es necesario que entre el enrutador *R* y la subred en donde se encuentra Nagios exista siempre un canal de Internet activo. Debido a esto y teniendo en cuenta la última característica mencionada en el apartado anterior, Nagios deberá monitorear tanto la disponibilidad del acceso a Internet en el enrutador *R* como la conexión VPN en el servidor *S*.

La lógica del monitoreo es muy simple en términos de definición de objetos en la configuración de Nagios.

Primer paso: se define un objeto de tipo 'host' (ver Listado 1) que apunta a la dirección IP pública que tiene signada el enrutador *R*. de esta manera se estaría definiendo el monitoreo del canal a Internet en el enrutador *R* (recordemos que si no hay canal de Internet en el enrutador *R* no puede existir la conexión VPN en el servidor *S*).

En la línea 2 se hace uso de una plantilla llamada 'generic-host' que define las propiedades básicas de este tipo de host.

En la línea 3 se le indica a Nagios cual será el nombre de este objeto, esta propiedad identifica al objeto dentro del sistema de monitoreo.

En la línea 4 se le define un alias que es usado para mostrar más amigablemente el nombre del objeto monitoreado en la interfase de administración de Nagios.



Figura 1: Escenario de monitoreo

```
1: define host{
2:     use         generic-host
3:     host_name   Enrutador-R
4:     alias       Enrutador R
5:     address     200.1.1.1
6: }
```

Listado 1: Definición del objeto para el enrutador R¹

¹ Las líneas están numeradas solo con propósitos ilustrativos. No incluya los números de las líneas en la definición del objeto en Nagios.

En la línea 5 (ver Listado 1) se le indica a Nagios cual es la dirección IP del objeto que se está definiendo. Nagios utiliza esta propiedad para hacer 'ping' a esta dirección y monitorear la disponibilidad de este objeto.

Segundo paso: se define un objeto de tipo 'host' (ver Listado 2) que apunta a la dirección IP privada que tiene asignada la conexión VPN en el servidor S (las conexiones VPN del tipo PPTP crean una interfase virtual PPP con una dirección IP en el host que inicia la conexión VPN).

La definición de este objeto tiene la particularidad de que depende del primer objeto, ya que el primer objeto es quién dispone del recurso de conexión a Internet. Esto es bien importante a la hora de recibir las notificaciones. Esta definición de dependencia le indica a Nagios la misma lógica que se está planteando aquí, es decir, Nagios sabrá que si no hay canal de Internet en el enrutador R no puede existir conexión VPN en el servidor S y dejará de notificar este problema.

Las líneas 2 a la 4 son análogas a la definición del primer objeto de listado 1, la línea 5 indica la dirección IP de este objeto, esta dirección IP esta asignada en la interfase virtual PPP que crea la conexión VPN. El servidor de Nagios se encuentra en el mismo segmento de subred y es capaz de alcanzar esta dirección IP sin inconvenientes. En la línea 6 se está definiendo la dependencia de la conexión VPN con

la disponibilidad del canal de Internet en el enrutador R.

```
1: define host{
2:   use         generic-host
3:   host_name   VPN-en-servidor-S
4:   alias       VPN en servidor S
5:   address     192.168.1.20
6:   parents     Enrutador-R
7: }
```

Listado 2: Definición del objeto para el servidor S

Manejadores de eventos

Un manejador de evento es un componente de software que se ejecuta cuando se detecta un cambio en el estado de un host o un servicio que se está monitoreando. Por ejemplo, si la conexión VPN cambia de su estado conectado a desconectado (sin que el canal de Internet del enrutador R haya cambiado a un estado desconectado), es posible disparar la ejecución de un componente de software que levante la conexión VPN en el servidor S. Para una mejor comprensión de la lógica con la que deben construirse los manejadores de eventos, se recomienda visitar el enlace relacionado en el apartado de referencias.

Para este caso se ha diseñado el manejador de evento detallado en listado 3, el cual es capaz de iniciar nuevamente la conexión VPN por solicitud del componente de monitoreo de Nagios. Cabe apuntar que este manejador de evento estará instalado en el servidor S a través del componente 'nrpe_nt' del que se hablará mas adelante.

Debido al alcance de este artículo explicar cada línea de código del manejador de evento descrito en listado 3 queda imposible, pero se puede mencionar que está escrito en Visual Basic Script y hace uso del comando 'rasdial.exe' disponible en el sistema operativo Windows para desconectar o conectar la conexión VPN a través de la línea de comandos.

Los argumentos que recibe este manejador de evento en la línea de comandos son los siguientes en el mismo orden listado a continuación:

1. **Nombre de la conexión VPN:** nombre con el que se creó la conexión VPN en el sistema operativo.
2. **Usuario:** nombre del usuario utilizado en la conexión VPN para autenticarse.
3. **Password:** contraseña del usuario utilizada en la conexión VPN para autenticarse.
4. **Intento:** número de chequeo en el que debe levantarse la conexión VPN.
5. **Estado del host:** este parámetro lo pasa Nagios en una macro denominada \$HOSTSTATE\$.
6. **Tipo de estado del host:** ídem al anterior en una macro denominada \$HOSTSTATETYPE\$.
7. **Número de intento:** número que indica cuantas veces Nagios ha revisado el estado del objeto desde que se detectó un cambio

en el mismo. Este parámetro lo pasa Nagios en una macro denominada \$HOSTATTEMP\$. El valor de este parámetro es comparado con el valor del cuarto parámetro para determinar si la conexión debe ser levantada.

Para probar el código del manejador de evento, debe copiarse a un archivo de texto y guardarse con extensión '.vbs', luego debe irse al símbolo del sistema y ejecutar el script de la siguiente manera:

```
cscript //nologo UpPPP.vbs <arg1> <arg2>
<arg3> <arg4> <arg5> <arg6> <arg7>
```

Ejemplos:

```
cscript //nologo UpPPP.vbs miconn pepito
passpepito 2 DOWN SOFT 2
```

```
cscript //nologo UpPPP.vbs miconn pepito
passpepito 2 DOWN HARD 2
```

Lo que está en mayúsculas corresponde al valor de las macros \$HOSTSTATE\$, \$HOSTSTATETYPE\$ y \$HOSTATTEMP\$ y para las pruebas estos argumentos deben pasarse en mayúsculas ya que al interior del código del manejador de evento, éstos se comparan en mayúsculas.

El componente NRPE_NT

Para instalar el manejador del evento es necesario descargar e instalar el componente 'nrpe_nt' sobre el servidor S. Para instalar el componente 'nrpe_nt' se recomienda leer el documento INSTALL que contiene el paquete descargado.

```

1  ' Copyright 2008 Oteon S.A.
2  '
3  ' This program is free software; you can redistribute it and/or
4  ' modify it under the terms of the GNU General Public License
5  ' as published by the Free Software Foundation; version 2
6  ' of the License.
7  '
8  ' This program is distributed in the hope that it will be useful,
9  ' but WITHOUT ANY WARRANTY; without even the implied warranty of
10 ' MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
11 ' GNU General Public License for more details.
12 '
13 'Autor: Juan Felipe Muñoz Fernandez [jmunoz@oteon.com]
14 'Dic 2008
15
16 Option Explicit
17
18 Const RASDIAL = "C:\WINDOWS\system32\rasdiag.exe"
19
20 Dim objShell, nRetorno, HostState, HostStateType, HostAttemp
21 Dim ConexionVPN, Attemp, Usuario, Password
22 Dim objExec, strConexionesActivas, ExpReg
23
24 Set objShell = CreateObject("WScript.Shell")
25
26 Function EstaConectado()
27     Dim Ret
28     Set ExpReg = new RegExp
29
30     Set objExec = objShell.Exec(RASDIAL)
31     strConexionesActivas = objExec.StdOut.ReadAll
32
33     ExpReg.Pattern = ConexionVPN
34     ExpReg.IgnoreCase = True
35
36     If ExpReg.Test(strConexionesActivas) Then
37         Ret = True
38     Else
39         Ret = False
40     End If
41
42     EstaConectado = Ret
43 End Function
44
45 Function DesconectarConectar()
46     If EstaConectado() Then
47         nRetorno = objShell.Run(RASDIAL & " " & ConexionVPN & " /DISCONNECT",0,True)
48         Wscript.Sleep 5000
49     End If
50     nRetorno = objShell.Run(RASDIAL & " " & ConexionVPN & " " & Usuario & " " &
51 Password,0,True)
52 End Function
53
54 ConexionVPN = WScript.Arguments(0)
55 Usuario = WScript.Arguments(1)
56 Password = WScript.Arguments(2)
57 Attemp = WScript.Arguments(3)
58 HostState = WScript.Arguments(4)
59 HostStateType = WScript.Arguments(5)
60 HostAttemp = WScript.Arguments(6)
61
62 If Trim(HostState) = "DOWN" Then
63
64     If Trim(HostStateType) = "SOFT" Then
65
66         If HostAttemp = Attemp Then
67             DesconectarConectar()
68         End If
69

```

```

70         End If
71
72         If Trim(HostStateType) = "HARD" Then
73             DesconectarConectar()
74         End If
75     End If
76     WScript.Echo "OK"
77     WScript.Quit(0)

```

Listado 3: Código del manejador del evento UpPPP.vbs

El componente 'nrpe_nt' depende de un archivo de configuración ubicado en el directorio 'bin'. A este archivo de nombre 'nrpe.cfg' deben modificársele las siguientes líneas por los valores ilustrados a continuación:

```

1: allowed_hosts=
2: dont_blame_nrpe=1
3: command_timeout=60

```

Listado 4: Directivas que cambian en el archivo 'nrpe_nt.cfg'

En la línea número 1 debe especificarse la dirección IP pública de donde proviene la conexión de Nagios.

En la línea número 2 debe habilitarse la posibilidad de que los comandos definidos puedan recibir argumentos.

En la línea número 3 simplemente se modifica el valor de espera para terminar la ejecución de los comandos cuando éstos tardan mucho en retornar.

Guarde los cambios en el archivo 'nrpe_nt.cfg' y reinicie el servicio de la siguiente manera en el símbolo del sistema:

```
net start nrpe_nt
```

Es buena práctica revisar el archivo 'nrpe_nt.log' ubicado en el directorio 'bin' para constatar de que no existen errores en el archivo de configuración 'nrpe_nt.cfg'.

También es importante mencionar que el servicio 'nrpe_nt' escucha el puerto 5666 en TCP, de tal manera que el administrador del sistema debe permitir las conexiones entrantes hacia este puerto en el enrutador R apuntando hacia el servidor S.

Instalación del manejador de evento

Una vez instalado y probado el componente 'nrpe_nt' en el servidor S, el primer paso es definir el comando en el archivo 'nrpe.cfg' como se indica en listado 5.

No olvide guardar los cambios en el archivo 'nrpe_nt.cfg' y reiniciar el servicio.

El segundo paso consiste en definir el mismo comando en el servidor Nagios de manera casi igual. Las líneas de listado 6 definen el comando para el manejador del evento en Nagios.

```
1: command[up_ppp]=cscript.exe //nologo //T:60 C:\nrpe_nt\UpPPP.vbs $ARG1$
   $ARG2$ $ARG3$ $ARG4$ $ARG5$ $ARG6$ $ARG7$
```

Listado 5: Definición el comando 'up_ppp' en el servicio 'nrpe_nt'

```
1: define command{
2:     command_name up_ppp
3:     command_line $USER1$/check_nrpe -t 60 -H $ARG1$ -c up_ppp -a $ARG2$
   $ARG3$ $ARG4$ $ARG5$ $HOSTSTATE$ $HOSTSTATETYPE$ $HOSTATTEMPT$
4: }
```

Listado 6: Definición del comando 'up_ppp' en el servidor Nagios

En la línea 2 del Listado 6 se está definiendo el nombre del comando para el manejador del evento.

En la línea 3 la macro \$USER1\$ tiene el valor '/usr/local/nagios/libexec' que es donde residen los plugins con los que Nagios ejecuta los monitoreos. El argumento \$ARG1\$ deberá valorarse con la dirección IP pública en donde se encuentra el componente 'nrpe_nt' a la escucha a la hora de su ejecución. Los demás argumentos y macros después del modificador '-a' corresponden con los argumentos que espera el script 'UpPPP.vbs'. Note como los tres últimos parámetros que espera el script 'UpPPP.vbs' corresponden a las macros antes mencionadas.

Si se desea comprobar el funcionamiento del comando anteriormente definido, se puede

ejecutar a mano, en la línea de comandos del servidor Nagios, la siguiente orden:

```
/usr/local/nagios/libexec/chec_nrpe -t 60
-H 200.1.1.1 -c up_ppp -a miconn pepito
passpepito 2 DOWN SOFT 2
```

Si previamente se ha publicado el puerto 5666 en TCP hacia el servidor S en el enrutador R, la orden llegará al servicio 'nrpe_nt' que se ejecuta en el servidor S y éste disparará la ejecución del comando 'up_ppp' con los parámetros que está recibiendo desde el servidor Nagios.

Para finalizar, faltaría ajustar la definición del objeto 'VPN-en-servidor-S' realizada en el Listado 2, para agregar el manejador del evento.

```
1: define host{
2:     use generic-host
3:     host_name VPN-en-servidor-S
4:     alias VPN en servidor S
5:     address 192.168.1.20
6:     parents Enrutador-R
7:     event_handler up_ppp!200.1.1.1!miconn!pepito!passpepito!2
8: }
```

Listado 7: Manejador de evento en la definición del objeto ilustrado en Listado 2

En la línea 7 se está definiendo el manejador del evento, la palabra 'up_ppp' corresponde al nombre del comando que se especificó con 'command_name'.

La dirección 200.1.1.1 es la dirección IP que tiene asignado el enrutador R y por donde se llega hasta el servidor S. La palabra 'miconn' corresponde al nombre de la conexión VPN en el servidor S.

Los siguientes dos parámetros corresponden al usuario y al password usados en la autenticación de la conexión VPN. El último parámetro '2' le dice al plugin 'UpPPP.vbs' que si la conexión VPN se encuentra desconectada y ya es

la segunda revisión que hace Nagios sobre el estado de la conexión luego de desconectarse, entonces es hora de reconectar la conexión VPN. El lector inquieto se preguntará ¿y donde están los argumentos restantes que espera el script 'UpPPP.vbs?', la respuesta está en la definición del comando 'up_ppp' del Listado 6.

Nagios puede llenar los valores de las macros \$HOSTSTATE\$, \$HOSTSTATETYPE\$ y \$HOSTATTEMPT\$ al momento en que ejecuta el manejador del evento. La macros anteriores pueden tomar los siguientes valores:

\$HOSTSTATE\$	Una cadena de caracteres que indica el estado actual del host: "UP", "DOWN" ó "UNREACHABLE"
\$HOSTSTATETYPE\$	Una cadena de caracteres que indica el tipo de estado en el que se encuentra el host actual: "SOFT" ó "HARD"
\$HOSTATTEMPT\$	El número de chequeos que lleva Nagios hacia el host luego de que se detectó un cambio en el estado del mismo.

Tabla 1: Valores de las macros usadas en el manejador de evento

Aplicaciones reales

La aplicación de este manejador de evento mejoró sustancialmente la disponibilidad de una conexión VPN en un entorno real de comunicaciones. Las siguientes imágenes ilustran las gráficas de disponibilidad de la conexión VPN sin el manejador del evento y la disponibilidad de la conexión con el manejador del evento. Teniendo en cuenta que se trata de una conexión

VPN que debe permanecer conectada las 24 horas.

En la figura 2, la gráfica ilustra como la mayoría del tiempo la conexión VPN permanecía desconectada (franja de color rojo).

La implementación del manejador del evento se comenzó a usar desde el 15 de enero de 2009 y para la fecha de finalización de este artículo, la mejora era sustancial en solo 24 horas de aplicación.

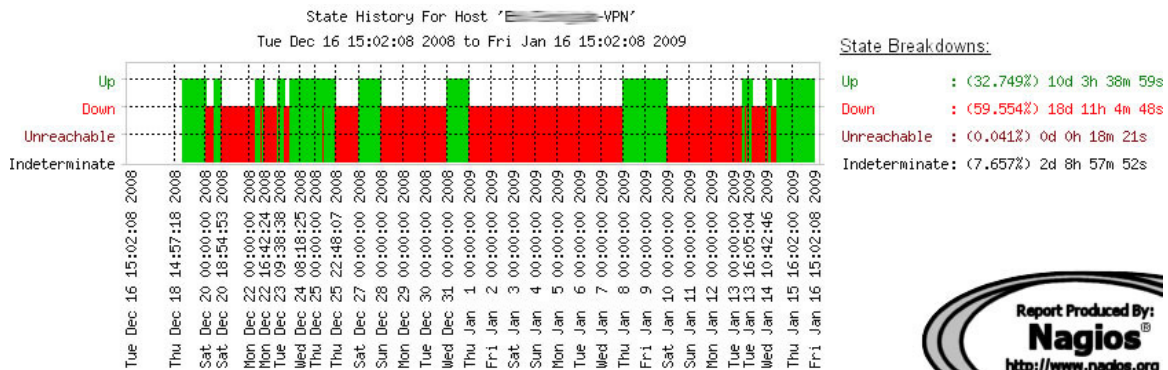


Figura 2: Reporte de disponibilidad de la conexión sin el manejador de evento

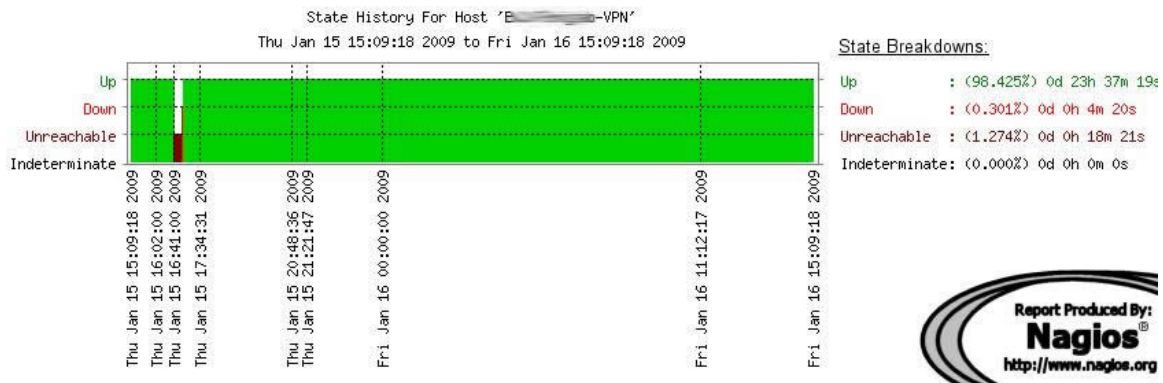


Figura 3: Reporte de disponibilidad de la conexión con el manejador de evento

Referencias

- Documentación de los manejadores de eventos en Nagios (inglés): http://nagios.sourceforge.net/docs/3_0/eventhandlers.html
- Comprender como funciona el componente NRPE_NT: http://nagios.sourceforge.net/docs/3_0/addons.html#nrpe
- Descargar del componente NRPE_NT: <http://sourceforge.net/projects/nrpen>

El autor

Juan Felipe Muñoz Fernández es Tecnólogo en Sistematización de Datos del Politécnico Colombiano Jaime Isaza Cadavid y actualmente cursa su último semestre de Ingeniería Informática en la misma institución. Se desempeña como gerente de la empresa Oteon S.A. liderando proyectos en las áreas de infraestructura informática y de telecomunicaciones en diferentes industrias del departamento de Antioquia, Colombia.