



Sistemas de Monitorización

José Ángel Muñoz Martínez

Hace unos días un amigo experto en redes que trabaja en una gran empresa de telecomunicaciones, me expresó una de las preocupaciones más comunes con las que me he encontrado durante mis últimos años de trabajo. Cómo gestionar y monitorizar cada uno de los dispositivos incluidos en cualquier red, sea del tamaño que sea, controlando el rendimiento desde un simple ordenador con un router de banda ancha a una red de más de 10000 equipos y servicios.

linux@software.com.pl

Desde mi punto de vista hay tres valores técnicos fundamentales a la hora de elegir un sistema de monitorización adecuado para nuestro entorno, independientemente del precio o facilidad de instalación y de utilización, los cuales también serán tenidos en cuenta. Estos tres valores son:

- Forma de presentar los datos, las alarmas y gráficos para su estudio. Estos han de ser lo más eficientes posible y ofrecer una idea de los problemas de un solo vistazo.
- Ubicación, distancia entre equipos y tipo de conexión (velocidad y rendimiento). A más distancia y cantidad de equipos, así como con conexiones lentas, es conveniente utilizar agentes locales que reporten a un servidor central.
- Sistemas Operativos que se monitorizarán. Es más sencillo monitorizar únicamente máquinas basadas en Linux, aunque en mi último proyecto era fundamental hacerlo con Windows y conveniente utilizar WMI, (además de SNMP, protocolo utilizado por todos los sistemas descritos en este artículo).

El Típico Tópico – ¿Lo más caro es siempre lo mejor?

La respuesta es, rotundamente, no. Lo más caro no siempre es lo mejor (al menos en lo que a Sistemas de Monitorización Remota se refiere). Según mi experiencia, todos los sistemas presentados en este artículo son, por supuesto, Open Source, totalmente personalizables, de fácil instalación y configuración. Poseen un rendimiento enviable, dejando a los sistemas comerciales en un segundo plano por su poca flexibilidad y su escaso abanico de configuraciones.

Los Sistemas de Monitorización instalados en Linux, nos permitirán también ahorrar en hardware, ya que con un simple Pentium IV, 512 MB de RAM y Ubuntu 7.10 Server instalado, os puedo prometer que nuestro servidor será capaz de recoger información de más de 100 dispositivos ubicados en cualquier lugar del mundo.

Sistemas de Monitorización Remota

A continuación se detallan las características de los paquetes de software Open Source de monitorización remota mejor considerados por la comunidad Linux:



Nagios

Originalmente llamado NetSaint (años 1999 a 2001), Nagios (www.nagios.org) es un monitor de servidores y aplicaciones diseñado originalmente para informar de problemas de una forma proactiva, reportándolos vía e-mail, SMS, mensaje instantáneo y/o solucionándolos automáticamente antes de que el cliente o usuario final pueda darse cuenta de los mismos. Probablemente Nagios haya sido el padre de todos los sistemas de Monitorización bajo Linux, pudiendo tanto su servidor como los agentes, ejecutarse únicamente desde este entorno. Estos agentes, permiten monitorizar equipos instalados en cualquier lugar con cualquier tipo de conexión. Ellos son los que se encargan de recoger los datos y enviarlos a un servidor central.

La potencia de Nagios viene dada por un sistema de *plugins* externos, encargados de enviar la información requerida. Estos *plugins* pueden programarse de una forma sencilla en bash o en perl, dando una potencia y flexibilidad sin igual a la herramienta.

Aunque la configuración es poco amigable, está basado en Web, donde podremos consultar el estado actual de nuestros equipos, históricos e informes. No tiene un sistema de gráficos que muestre de forma clara un historial de rendimiento (CPU, Memoria) o tráfico, pero su sistema de alarmas es perfecto para, de un vistazo, ver qué está fallando dentro de nuestra red. Os puedo prometer que el sistema de alarmas de Nagios puede llegar a convertirse en un problema y un caos si se organizase de forma incorrecta, dando lugar a que la herramienta deje de cumplir su principal función proactiva.

Centreon será el compañero de viaje perfecto para Nagios, ofreciéndole todo aquello de lo que no puede presumir y mejorando notablemente casi todas sus deficiencias (soporte MySQL, gráficos, configuración por web, una librería de plugins para chequeo SNMP y un largo etcétera).

Zabbix

Zabbix (nacido en el año 2001) (www.zabbix.com) es una solución de monitorización 24x7 de bajo coste, capaz de recoger datos de cualquier aplicación o servidor, con un sistema de envío y captura de datos a través de sus agentes. Es importante señalar que Zabbix es capaz de monitorizar, además, equipos Windows, AIX, FreeBSD, HP-UX, OpenBSD y Solaris *desde dentro* y personalizar sus agentes con parámetros definidos por el usuario, no tan potentes como los plugins de Nagios pero suficientes para gestionar cualquier equipo.

La ventaja de Zabbix reside en su interfaz web soportado por una base de datos SQL (yo siempre recomiendo MySQL, aunque funciona perfectamente con PostgreSQL o SQLite), donde la configuración se convierte en un juego de niños y donde cualquier persona con algún conocimiento de informática puede gestionar sin problemas esta potente herramienta.

Posee un sistema proactivo (Acciones) que permite solucionar automáticamente los problemas, un sistema de monitorización, alertas y visualización de gráficos que no tienen otros sistemas de monitorización, ni siquiera los mejores programas comerciales, haciendo de Zabbix, si no el mejor, uno de los mejores sistemas de Monitorización Remota.

Pandora FMS

Relativamente joven (año 2005), Pandora FMS - Free Monitoring System (pandora.sourceforge.net) permite, a través de agentes analizar el estado y rendimiento de los diferentes parámetros ofrecidos por cualquier plataforma del mercado (Linux, Solaris, MS Windows, AIX y otros).

Todas las comunicaciones se realizan a través de SSH, FTP, NFS o un contenedor XML para transportar los datos que se guardarán en una base MySQL de un Servidor Central, el cual, será el encargado de recopilar cada uno de esos datos y mostrarlos en una magnífica interfaz web.

Pandora FMS, aún con su complicada instalación, está condenado a ser uno de los mejores sistemas de Monitorización Remota junto con Zabbix y Nagios presumiendo de las mejores características de ambos y añadiendo una flexibilidad propia de los mejores.

Hay que recalcar que Pandora FMS permite controlar sistemas Windows a través de WMI, ideal para redes en compañías con instalaciones mixtas Linux/Windows.

Zenoss

Zenoss (año 2006) (www.zenoss.com), es ese software que sorprende por su capacidad e interfaz simplemente echando un vistazo a su página web. La razón por la que su potencial ha sido tan visible de una forma tan rápida ha sido, como siempre, el dinero, habiendo apostado por él varias empresas estadounidenses.

Funciona sobre Linux, FreeBSD y como novedad, sobre Mac OS X y VMWare Player, pudiendo ejecutarse de esta manera sobre Linux o Windows de la forma más sencilla posible.

Zenoss se distribuye en rpm para su instalación automática bajo RedHat, CentOS y Fedora Core 6. Para Suse, Debian, Ubuntu, FreeBSD, Gentoo, Solaris 10 y Mac OS X, se distribuyen los fuentes para ser compilados en cada uno de los sistemas mencionados, dificultando de esta manera la instalación.

Tiene una interfaz similar a la de Nagios. De hecho, puede importar sus plugins y, aunque su configuración no es todo lo sencilla que debería, es capaz de detectar los equipos de nuestra red automáticamente utilizando (además de SNMP) SSH. Zenoss no necesita agentes en las máquinas remotas, ya que con SSH puede ejecutar de forma segura cualquier comando que deseemos para extraer todo tipo de información. Para monitorizar máquinas Windows, utiliza un binario que conecta usando WMI para modelar y monitorizar sus servicios.

Por último, mencionar que el sistema de gráficos que utiliza es similar (si no el mismo) al de MRTG, mencionado más adelante. Más que suficiente para un sistema

Listado 1. Archivo cgi.cfg Original

```
authorized_for_system_information=nagiosadmin authorized_for_configuration_
information=nagiosadmin authorized_for_system_commands=nagiosadmin
authorized_for_all_services=nagiosadmin authorized_for_all_
hosts=nagiosadmin #authorized_for_all_services=nagiosadmin,guest
#authorized_for_all_hosts=nagiosadmin,guest authorized_for_all_service_
commands=nagiosadmin authorized_for_all_host_commands=nagiosadmin
```

Listado 2. Archivo cgi.cfg Modificado

```
authorized_for_system_information=universo authorized_for_configuration_
information=universo authorized_for_system_commands=universo
authorized_for_all_services=universo authorized_for_all_hosts=universo
#authorized_for_all_services=universo,guest #authorized_for_all_
hosts=universo,guest authorized_for_all_service_commands=universo
authorized_for_all_host_commands=universo
```

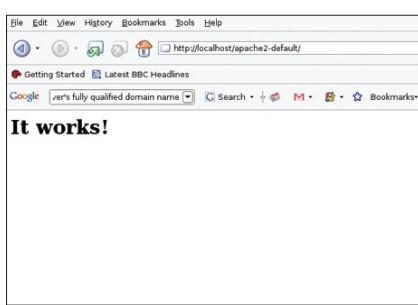


Figura 1. Pantalla Inicial de Apache



seguridad

Monitorización de sistemas

que acaba de nacer y que posee los mejores galardones del software Open Source.

Sistemas de monitorización local

A continuación vamos a detallar las características de los paquetes de software Open Source de monitorización local.

Nino

Nino (nino.sourceforge.net) permite gestionar routers, switches, servidores y aplicaciones. Está pensado para redes locales, ya que no dispone de agentes que puedan reportar a un servidor central. Basado en web, Nino no destaca por su interfaz, nada cómodo de gestionar. Utiliza SNMP, HTTP/TCP y WMI. Sus gráficos son suficientes para una pequeña red con máquinas Linux y Windows.

OpenNMS

OpenNMS (www.opennms.org) es experto en monitorizar equipos en una red local a través de SNMP con un sistema muy robusto de notificaciones. Es, junto con Nagios, el padre de la monitorización remota. Posee una buena interfaz web, aunque la evaluación de los datos mostrados se hace a veces complicada. Su demo online (demo.opennms.org/) nos puede dar una muy buena idea de su funcionamiento y evaluarlo sin necesidad de instalarlo en nuestro equipo.

MRTG

MRTG (The Multi Router Traffic Grapher) (www.mrtg.org). MRTG es la referencia para la monitorización y medición de carga en una red.

Escrito en Perl, puede trabajar en Linux, Unix, Windows Mac OS y Netware y utiliza SNMP para recoger los datos. Aunque su instalación y configuración no es muy intuitiva, su consumo de recursos es tan pequeño y su información tan valiosa que merece la pena instalarlo en cualquier máquina que nos encontremos por el camino.

He visto medir de casi todo con MRTG (Temperatura, Iluminación, Flujo de agua en un conducto, Señal Analógica de Televisión e incluso Ocupación en Hoteles). Muy recomendable: RRDtool como sistema de Base de Datos para MRTG y Cacti como interfaz web para RRDtool.

Instalación y configuración de Nagios

Aunque hay diferentes caminos para instalar y configurar Nagios (www.nagios.org) y, aunque en la documentación oficial avisan de la necesidad de compilar y de la complejidad de la instalación y configuración, a continuación se explica de la manera más sencilla y automática posible utilizando la distribución Ubuntu 7.10 Gutsy Gibbon. Toda la instalación se realizará desde una consola Linux, siendo efectiva tanto para la distribución Server (Servidor) como para la WorkStation (Estación de Trabajo).

Los archivos de configuración de ejemplo nos ayudarán a la hora de crear una configuración base y empezar a monitorizar nuestros equipos. Tras configurarlo, veremos la forma de crear nuestros propios plugins. Hay que recordar que en ellos reside el potencial y la gran capacidad de monitorización de Nagios, haciéndolo totalmente flexible.

Aunque requerimos únicamente un equipo en Linux instalado y una conexión TCP/IP, para visualizar el estado de nuestros equipos, necesitaremos apache2 instalado. Explicaremos su instalación de una forma sencilla y breve para el correcto funcionamiento de la interfaz web de Nagios.

Una curiosidad que también veremos: Firefox dispone de una extensión que avisa de los problemas en nuestros equipos.

Instalación del Servidor de HTTP Apache 2

A la hora de instalar cualquier paquete en Debian y más concretamente en Ubuntu, nos damos cuenta del potencial de este sistema operativo en general y de Debian/Ubuntu en particular. Quiero decir esto porque al utilizar el gestor de paquetes apt veremos lo fácil que resulta la instalación. En el caso de apache ejecutaremos:

```
$ sudo apt-get install apache2
```

Figura 2. Pantalla Inicial de Nagios

Figura 3. Configuración Predeterminada de Nagios



El gestor apt descargará, instalará y configurará automáticamente nuestro servidor HTTP, así como cualquier otro paquete necesario si lo necesitara. En el caso de que apache2 estuviera ya instalado en nuestro sistema intentaría actualizarlo. Si ya tuviéramos la última versión, nos mostrará el mensaje *apache2 is already the newest version* confirmando que ya disponemos de la versión más actualizada.

Para probar su correcto funcionamiento, abriremos nuestro navegador (yo uso Firefox) y escribiremos en la barra de direcciones la URL <http://localhost/apache2-default/>. Si todo ha ido bien, nos mostrará el mensaje *It works* (¡Funciona!), ver Figura 1.

Si tras la instalación y en cada inicio del demonio de apache se nos mostrara el error: *apache2: Could not reliably determine the server's fully qualified domain name*

Podremos evitarlo editando el archivo httpd.conf con nuestro editor de textos favorito:

```
$ sudo vi /etc/apache2/httpd.conf
```

Y añadiremos la línea:

```
ServerName localhost
```

A partir de aquí, hay un mundo de parámetros para nuestro servidor, el cual, si lo vais a exponer a Internet, es muy recomendable configurarlo para que utilice HTTP Seguro (puerto 443).

Instalación de Nagios

Una vez instalado nuestro servidor web, instalaremos nagios utilizando el mismo proceso utilizado para apache. Desde una consola Linux, ejecutaremos el comando:

```
$ sudo apt-get install nagios2 nagios-plugins nagios-images nagios-nrpe-plugin
```

Una vez confirmado que queremos iniciar la instalación, comprobaréis que nuestro querido Ubuntu, de una forma totalmente automática, nos instalará los paquetes necesarios y sus dependencias para que nuestra instalación termine de la forma más exitosa, dejando toda la preocupación a nuestro gestor de paquetes apt.

Configuración Inicial

En este punto hay que dejar la impaciencia a un lado, ya que nuestro Nagios, todavía requiere de algunos ajustes básicos antes de empezar a funcionar. Al igual que la

recomendación de mostrar nuestro servidor apache de una forma segura con https, el primero de los ajustes es cambiar, por la misma razón, el nombre de nuestro Administrador de Nagios que por defecto es *nagiosadmin*. Para ello editaremos el archivo cgi.cfg de Nagios:

```
$ sudo vi /etc/nagios2/cgi.cfg
```

Y cambiaremos el usuario nagiosadmin en las líneas que se indican en el Listado 1 (incluidas las comentadas con '#' para que no se nos olvide en un futuro) cgi.cfg con nuestro usuario Administrador de Nagios. Utilizaremos el nombre de usuario *universo* como muestra el Listado 2. Una vez modificado nuestro Administrador, le asignaremos una contraseña con el comando htpasswd. Este comando se utiliza para autenticación básica, donde el parámetro '-c' sirve para crear el archivo de contraseñas htpasswd.users.

```
$ sudo htpasswd -c /etc/nagios2/htpasswd.users universo
```

De nuevo y por seguridad, es recomendable cambiar el nombre del archivo de contraseñas por uno no estándar. Esta operación puede realizarse modificando el archivo /etc/nagios2/apache2.conf en la línea: AuthUserFile /etc/nagios2/htpasswd.users

Con nuestro usuario Administrador cambiado y definida su contraseña, ya podemos abrir nuestro navegador y acceder a nuestro portal de Nagios: <http://localhost/nagios2>

Listado 3. Archivo contacts.cfg

```
define contact{
    contact_name      administrador
    alias             Jose
    service_notification_period 24x7
    host_notification_period   24x7
    service_notification_options w,u,c,r
    host_notification_options  d,r
    service_notification_commands notify-by-email
    host_notification_commands host-notify-by-email
    email              josea.munoz@gmail.com
}
```

Listado 4. Archivo contactgroups.cfg

```
define contactgroup{
    contactgroup_name   admins
    alias               Administradores Nagios
    members             administrador
}
```

Una vez introducida correctamente la contraseña, se nos mostrará la pantalla principal, ver Figura 2.

Seleccionando la opción *Service Detail* veremos que automáticamente, Nagios ha comenzado a monitorizar nuestro servidor y nuestra puerta de enlace predeterminada, ver Figura 3.

Ya tenemos nuestro servidor Nagios funcionando con una configuración básica. Ahora configuraremos y añadiremos funcionalidades extras a nuestra instalación.

Configuración de Nagios

Como ya comentamos en el estudio inicial, y como el lector ha podido comprobar, uno de los peros de Nagios es su modo de configuración, que ha de ser (por ahora) editando sus archivos en una consola Linux.

Archivos de configuración principales

Son, los siguientes archivos los encargados de la configuración básica de Nagios:

- *apache2.cfg* – Contiene la configuración relativa al servidor web apache, una de esas configuraciones es la localización del archivo de autenticación.
- *cgi.cfg* – Archivo que contiene las directivas usadas por la interfaz web. Destacan los parámetros siguientes:
 - Grupos de usuarios que tendrán acceso a visualizar las diferentes secciones de Nagios2 en su interfaz web
 - Localización del archivo de configuración de Nagios2 (por defecto *nagios.cfg*)



Ubicación de los documentos HTML. Ubicación de los archivos multimedia de audio.

- *nagios.cfg* – Archivo de configuración principal para Nagios, donde se encuentran los principales parámetros. Entre otros:
 - Ubicación de los principales archivos para el funcionamiento de Nagios: Archivos log, lock y temp.
 - Método de rotación de archivos
 - Ubicación de los archivos de configuración
- *resources.cfg* – Indica dónde se encuentran los plugins de Nagios.

Listado 5. Archivo hosts.cfg

```
# Para chequear si la línea de salida a Internet es correcta, es útil hacer ping a una URL externa.

define host{
  host_name google
  alias Conexion Internet
  address www.google.com
  use generic-host
}

define host{
  host_name miwangateway
  alias ISP Gateway
  address 217.126.169.207
  parents google
  use generic-host
}

define host{
  host_name milangateway
  alias LAN Internet Gateway
  address 192.168.0.1
  parents miwangateway
  use generic-host
}

define host{
  host_name nagios
  alias Servidor Nagios
  address 192.168.0.18
  parents milangateway
  use generic-host
}

define host{
  host_name windows
  alias Windows Server
  address 192.168.0.9
  parents milangateway
  use generic-host
}
```

Monitorizar nuestra red

Configurar nuestros monitores es a partir de este momento una tarea digamos *pesada*, ya que es muy importante entender cada uno de los archivos de configuración y leer, sobre todo leer, la documentación de Nagios, así como buscar ejemplos de otros usuarios.

Para entender el funcionamiento de algunos de los *archivos.cfg*, es preferible crearlos desde cero, a copiarlos de alguna plantilla.

Empezaremos modificando la ubicación de nuestros archivos. Para ser un poco organizados, mantendremos las configuraciones existentes de */etc/nagios2/conf.d* y los utilizaremos como referencia. Luego crearemos un nuevo directorio */etc/nagios/config* para ubicar nuestros propios archivos de configuración.
\$ sudo mkdir /etc/nagios2/config. Para hacer efectivo el cambio, deberemos editar el archivo */etc/nagios2/nagios.cfg*. Buscaremos la línea:
cfg_dir=/etc/nagios2/conf.d

Listado 6. Archivo hostgroups.cfg

```
# HostGroup Comodín

define hostgroup {
  hostgroup_name todos
  alias Todos los Servidores
  members *
}

# Lista de Servidores
define hostgroup {
  hostgroup_name servidores
  alias Servidores
  members nagios, windows
}

# Lista de Servidores Web
define hostgroup {
  hostgroup_name servidores-http
  alias Servidores HTTP
  members nagios, windows
}

# Listas de Servidores SSH
define hostgroup {
  hostgroup_name servidores-ssh
  alias Servidores SSH
  members nagios
}

define hostgroup {
  hostgroup_name dispositivos-ping
  alias Servidores para Ping
  members google, miwangateway, milangateway, nagios, windows
}
```

Y la cambiaremos por *cfg_dir=/etc/nagios2/config*

Ahora que ya disponemos de nuestro propio directorio de configuración, empezaremos a crear, dentro del directorio */etc/nagios2/config* los archivos necesarios para nuestro entorno. Abajo enumeramos los que necesitaremos:

- *contacts.cfg*: donde se definen los usuarios que recibirán las alertas en caso de problemas.
- *contactgroups.cfg*: organiza los contactos en diferentes grupos.
- *hosts.cfg*: aquí se encuentran los dispositivos que se monitorizarán.
- *hostgroups.cfg*: organizaremos los dispositivos del archivo hosts.cfg en grupos (Servidores, Routers, Estaciones de Trabajo).
- *services.cfg*: define los servicios que se monitorizarán para cada dispositivo.
- *templates.cfg*: contiene plantillas personalizadas de comandos que se usarán con frecuencia en nuestras configuraciones.
- *timeperiods.cfg*: en este archivo de configuración, ajustaremos los diferentes períodos de tiempo para usarlos en nuestras configuraciones (Horas de Trabajo, Vacaciones, etc.).

Toda esta configuración, podría organizarse en un único archivo, el cual, con una configuración extensa, será totalmente inmanejable. En nuestro ejemplo usaremos la siguiente configuración. Dos Ordenadores y un Router:

- 192.168.0.1: Router – Puerta de Enlace
- 192.168.0.18: Servidor de Monitorización, Web, FTP y Correo
- 192.168.0.9: Servidor Windows

contacts.cfg (Listado 3)

- service_notification_options: w,u,c,r,f,n
- w=alarma u=desconocido c=crítico r=recuperado f=inestable n=ninguno
- host_notification_options: d,u,r,f,n
- d=caído u=inoperativo r=recuperado f=inestable n=ninguno, contactgroups.cfg(Listado 4)

hosts.cfg (Listado 5)

- host_name: Nombre único identificador del dispositivo
- alias: Descripción corta
- address: Dirección IP o nombre de dominio del dispositivo



- parents: Lista separada por comas, donde se enumeran los *padres* del dispositivo, útil para mostrarlo de forma esquemática en el mapa de Nagios.
- use: El nombre de la plantilla que se usará.

hostgroups.cfg (Listado 6)

- hostgroup_name: Nombre único identificador del grupo de dispositivos
- alias: Descripción corta

- members: Miembros del grupo de dispositivos

services.cfg (Listado 7)

- host_name: El dispositivo que ejecuta el servicio.
- hostgroup_name: En vez del nombre de dispositivo, este parámetro puede usarse con el nombre de un grupo. En este caso, se chequearán todos los dispositivos del grupo.
- service_description: Descripción corta del servicio.
- check_command: Comando que se ejecutará para chequear el servicio. Cada comando necesitará un plugin disponible (que se encuentran en /usr/lib/nagios/plugins).
- use: Nombre de la plantilla que se usará
- notification_interval: Período de tiempo para reenviar un cambio de estado del dispositivo. Si se configura a 0, Nagios lo notificará sólo una vez.

Listado 7. Archivo services.cfg

```
## Servicios de Hostgroups ##

# Chequear que los servicios web funcionan
correctamente

define service {
    hostgroup_name      servidores-http
    service_description  HTTP
    check_command        check_http
    use                 generic-service
    notification_interval 0
}

# Chequear que los servicios ssh están ejecutándose
define service {
    hostgroup_name      servidores-ssh
    service_description  SSH
    check_command        check_ssh
    use                 generic-service
    notification_interval 0
}

# Chequear que los dispositivos ping están disponibles
define service {
    hostgroup_name      dispositivos-ping
    service_description  PING
    check_command        check_ping!100.0,20%!500.0,60%
    use                 generic-service
    notification_interval 0
}

## Servicios de dispositivos individuales ##

# Para monitorizar cómo monitorizar un servicio en un dispositivo,
los servicios ftp y de correo se ejecutarán
individualmente para cada dispositivo.

define service {
    host_name           nagios ; Recordad que este
    servidor es también FTP.
    service_description  FTP
    check_command        check_ftp
    use                 generic-service
}

notification_interval 0

}

}

# Servicios para el servidor de correo (chequear: pop,
imap, pops, imaps and smtp).
define service {
    host_name           nagios
    service_description  POP
    check_command        check_pop
    use                 generic-service
    notification_interval 0
}

define service {
    host_name           nagios
    service_description  IMAP
    check_command        check_imap
    use                 generic-service
    notification_interval 0
}

define service {
    host_name           nagios
    service_description  Secure POP
    check_command        check_spop
    use                 generic-service
    notification_interval 0
}

define service {
    host_name           nagios
    service_description  Secure IMAP
    check_command        check_simap
    use                 generic-service
    notification_interval 0
}

define service {
    host_name           nagios
    service_description  SMTP
    check_command        check_smtp
    use                 generic-service
    notification_interval 0
}
```



Para añadir nuevos servicios, podemos echar un vistazo en `/usr/lib/nagios/plugins` (directorio por defecto de los plugins de Nagios) y buscar el plugin necesario para el servicio que deseemos chequear. Si los que hay no cumplen nuestras expectativas tenemos dos opciones: Buscar en la red (un buen sitio para empezar es <http://nagiosplugins.org>) o crearlos nosotros mismos (veremos más adelante varios ejemplos). Una vez que tenemos el plugin, necesitamos ejecutarlo con `--help` para ver qué parámetros son necesarios para ejecutarlo. Nuestro ejemplo nos muestra algunos servicios básicos. Importante: cada parámetro enviado al comando ha de separarse de los otros con un símbolo de cierre de exclamación (!).

Para más información sobre configuración de Nagios visitad: http://nagios.sourceforge.net/docs/2_0/xodtemplate.html (En Inglés).

templates.cfg

En la instalación por defecto, encontraremos las plantillas `generic-host_nagios2.cfg` y `generic-service_nagios.cfg` que copiaremos a nuestro directorio de configuración.

```
$ sudo cp /etc/nagios2/conf.d/
generic-service_nagios2.cfg /etc/
nagios2/config/template-service.cfg
$ sudo cp /etc/nagios2/conf.d/
generic-host_nagios2.cfg /etc/
nagios2/config/template-host.cfg
```

timeperiods.cfg

Realizaremos la misma operación que con las plantillas, usando el archivo de la instalación de Nagios:

```
$ sudo cp /etc/nagios2/conf.d/
timeperiods_nagios2.cfg /etc/nagios2/
config/timeperiods.cfg
```

Listado 9. Ejemplo Plugin de Nagios

```
#!/bin/bash

grep pcmcia /proc/modules >/dev/
null
if [ $? -eq 0 ]
then
echo "Portátil"
exit 0
fi
echo "Estación de Trabajo"
exit 0 #Si quisieramos mostrar un
error de "módulo no encontrado"
pondremos exit 2
```

Comprobar nuestra configuración.

Para comprobar que nuestra configuración es correcta es necesario reiniciar el demonio de Nagios:

```
$ sudo /etc/init.d/nagios2 restart
```

Y comprobarla en nuestro navegador: `http://localhost/nagios2`. Seleccionando la opción `Service Detail` en el menú, veremos nuestros

servicios, ver Figura 4. Podremos hacer lo mismo con nuestros dispositivos seleccionando la opción `Host Detail` en el menú, ver Figura 5.

Plugins de Nagios

Como ya hemos comentado anteriormente, una de las razones del potencial de Nagios son sus plugins. Éstos son los que dan a esta herramienta la capacidad de medir y controlar

Listado 10. Instalación de Centreon

```
Where is installed Nagios ?
default to [/usr/local/nagios]:/usr/lib/cgi-bin/nagios2

Where is your nagios etc directory ?
default to [/usr/lib/cgi-bin/nagios2/etc]:/etc/nagios2

Where is your nagios var directory ?
default to [/usr/lib/cgi-bin/nagios2/var]:/var/lib/nagios2

Where is your nagios plugins (libexec) directory ?
default to [/usr/lib/cgi-bin/nagios2/libexec]:/usr/lib/nagios/plugins

Where is your nagios bin directory?
default to [/usr/lib/cgi-bin/nagios2/bin]:/usr/sbin

Where is your nagios image directory ?
default to [/usr/lib/cgi-bin/nagios2/share/images]:/usr/share/nagios2/
htdocs/images

Where do I install centreon ?
default to [/usr/local/centreon]:
Do you want me to create this directory [/usr/local/centreon]?[Y/n]y

Where is sudo configuration file?
default to [/etc/sudoers]: <enter>

Where is installed RRD perl modules [RRDs.pm] ?
default to [/usr/local/rrdtool/lib/perl]:/usr/lib/perl5

Where is rrdtool binary ?
default to [/usr/bin/rrdtool]: <enter>

Where is mail binary ?
default to [/usr/bin/mail]: <enter>

Where is PEAR Path ?
default to [/usr/share/pear]:/usr/share/php

Do you want to install Centreon Plugins ?
[y/n], default to [y]:y

Do you want to install Centreon Traps Plugins ?
[y/n], default to [y]: <enter>

Where is your SNMP configuration file?
default to [/etc/snmp/]: <enter>
```



cualquier dispositivo o servicio de una forma más o menos sencilla, con unas simples nociones de programación en bash o perl.

Para aquellos que no deseen perder demasiado tiempo programando, además de la ya comentada <http://nagiosplugins.org>, podemos visitar <http://www.nagiosexchange.org/>. Muy recomendable también instalar el juego de plugins para chequear dispositivos con el protocolo snmp activado. La instalación, como siempre, con:

```
$ sudo apt-get install nagios-snmp-
plugins
```

Mostraremos a continuación un ejemplo práctico de cómo crear nuestros propios plugins.

Ejemplo: Chequear un módulo instalado

Podremos utilizar este ejemplo para chequear el contenido de cualquier archivo de nuestro sistema. En nuestro ejemplo, lo utilizaremos para comprobar si un módulo está instalado y funcionando – Ver Listado 9. Ejemplo Plugin de Nagios.

En este caso chequearemos si existe un módulo llamado `pcmcia` en `/proc/modules`. Este módulo suele encontrarse únicamente en instalaciones de equipos portátiles, así que daremos por hecho que si no se encuentra, es una estación de trabajo fija.

Para implementarlo, primero deberemos crear nuestro plugin en `/usr/lib/nagios/plugins` y le daremos permisos de ejecución:

```
$ sudo vi /usr/lib/nagios/plugins/
check_portatil
$ sudo chmod 755 /usr/lib/nagios/
plugins/check_portatil
```

Lo agregaremos a la lista de plugins creando un nuevo archivo de configuración dentro del directorio de configuración de plugins de nagios ubicado en `/etc/nagios-plugins/config/`:

```
$ sudo vi /etc/nagios-plugins/config/
nios.cfg
```

donde añadiremos nuestro nuevo plugin y la forma de utilizarlo. Este plugin es informativo, así que no necesita parámetros.

```
define command{
    command_name  check_portatil
    command_line  /usr/lib/nagios/
plugins/check_portatil)
```

Y lo utilizaremos como `check_command` en nuestro `services.cfg`

```
define service
{
    host_name          nagios
    service_description Portatil
    check_command      check_portatil
    use                generic-service)
```

Una vez reiniciado Nagios con el comando:

```
$ sudo /etc/init.d/nagios2 restart
```

podremos comprobar el resultado en Figura 6.

Monitorizar Máquinas Remotas con Nagios

NRPE es un cliente de Nagios diseñado para permitir ejecutar plugins en máquinas Linux remotas. Su función principal es permitir a Nagios monitorizar recursos *locales* (carga de CPU o memoria) en esos dispositivos remotos. Un servidor NRPE ha de estar instalado para ejecutar los plugins y devolver la información al cliente.

Ésta operación puede realizarse de forma más segura y eficaz a través de SSH, pero también requiere de mucha carga de CPU, tanto en el servidor como en el cliente. Siendo NRPE

la solución para redes de cientos o miles de dispositivos. NRPE consta de dos piezas:

- Plugin `check_nrpe`, que reside en el servidor de Nagios.
- Demónio NRPE, ejecutándose en la máquina remota.
- Cuando Nagios necesite monitorizar un recurso de la máquina remota,
- Ejecutará el plugin `check_nrpe` indicando el servicio a chequear
- El plugin `check_nrpe` contactará con el demónio NRPE del equipo remoto sobre una conexión (opcional) SSL.
- El demonio ejecutará el plugin necesario para chequear el servicio o recurso que enviará el resultado de vuelta al plugin NRPE.
- El Plugin NRPE enviará la información al servidor de Nagios.

Nagios NSCA Nagios NSCA trabaja exactamente a la forma inversa a la que lo hace Nagios NRPE.

En este caso, es la máquina remota la que envía la información de estado al servidor de Nagios central de una manera segura. NSCA también consta de dos piezas:

- Demónio NSCA, el cual se ejecuta en el servidor Nagios. Se encarga de re-

Tabla 3. Estado_Plugin_Ejemplo_de Nagios XXXXXXXXXXXXXXXXX

Portail	OK	2008-01-20 11:12:45	Od 0h 0m 46s	¼	Estacion de Trabajo
---------	----	---------------------	--------------	---	---------------------

Tabla 1. Los parametros de la base de datos.

Root password for Mysql	Indicaremos la contraseña de root de nuestro servidor de bases de datos MySQL
Centreon Database Name	centreon
Centreon Data Storage Database Name	cds
Database Password	Escribiremos nuestra contraseña para la base de datos de Centreon
Confirm it	Confirmar la contraseña anterior
Database location (localhost if blank)	(Dejar en blanco si es local)
Nagios location (localhost if blank)	(Dejar en blanco si es local)
MySQL Client version (Password Haching Changes)	Por defecto [>=4.1 - PASSWORD()]

Tabla 2. XXXXXXXXXXXXXXXXX

Administrator login for Oreon	admin
Administrator password for Oreon	Introducir contraseña
Confirm Password	Confirmar contraseña
Administrator firstname for Oreon	Nombre del Administrador
Administrator lastname for Oreon	Apellido del Administrador
Administrator Email for Oreon	email del Administrador
Administrator language for Oreon	Idioma (Sólo inglés – en y francés – fr)



coger los resultados de los dispositivos remotos (los cuales son enviados con el programa *send_nsca* explicado a continuación). Una vez recibidos los datos del cliente remoto, el demonio los validará de una forma básica. Esta operación implica desencriptar los datos con una contraseña guardada en el archivo de configuración *nsca.cfg*. Si son correctos (fueron encriptados por el programa *send_nsca* usando la misma contraseña), el demonio procesará las entradas y las confirmará a Nagios para que procese los resultados del dispositivo o servicio.

El demonio NSCA ha de tener los suficientes permisos de escritura en el archivo de coman-

dos de Nagios. Además, Nagios procesará únicamente aquellos datos encontrados en ese archivo y que hayan sido definidos en el archivo de configuración de dispositivos *hosts.cfg*.

Cliente *send_nsca* el cual envía la información monitorizada desde un dispositivo remoto al demonio nsca en el servidor central de Nagios.

Documentación de Nagios

Sólo un apunte muy importante. Nagios no termina, ni siquiera empieza en este artículo.

Es fundamental leer y volver a leer cualquier documentación de Nagios que nos encontremos en la red. Nagios puede

personalizarse de tal manera, que podríamos crear con su base, nuestro propio sistema de monitorización.

Nagios tiene la posibilidad de guardar los datos (información de estado, histórico, notificaciones, etc) en una base de datos MySQL.

Al igual que NSCA y NRPE consiste en un módulo cliente y un demonio. Al ser una opción experimental, no se ha analizado en este artículo.

Recordad que si queremos enviar correos necesitaremos un Servidor SMTP, ya que Nagios es capaz de avisarnos por correo, o mensajes SMS en caso de problemas.

Centreon: mejorando Nagios

Centreon u Oreon (www.centreon.com), es una herramienta de monitorización de redes, sistemas y servicios basada en Nagios, ofreciéndole una nueva y diferente interfaz web así como nuevas funcionalidades, haciendo más eficiente la monitorización y consiguiendo que la información sea más legible y más fácil de gestionar.

A continuación se explica su instalación, dando por supuesto que tenemos nuestro Nagios anterior configurado y funcionando correctamente.

Instalación de MySQL

Centreon necesita el gestor de bases de datos MySQL. Para instalarlo ejecutaremos:

```
$ sudo apt-get install mysql-server
```

PHP5

Utilizaremos la versión 5 de PHP. Además junto con PHP5, instalaremos el módulo para gestionar nuestras bases de datos en MySQL.

```
$ sudo apt-get install php5
$ sudo apt-get install php-db php-date
$ sudo apt-get install php5-gd php5-mysql php5-snmp php5-ldap
$ sudo apt-get install php-mail
php-mail-mime php-net-smtp php-net-socket
$ sudo apt-get install php5-xmlrpc
$ sudo apt-get install phpmyadmin
```

RRDtool

Es el sistema de gráficos utilizado por Centreon y utiliza perl para generarlo.

```
$ sudo apt-get install rrdtool
librrds-perl libconfig-inifiles-perl
```

The screenshot shows the Nagios configuration interface. On the left is a sidebar with various monitoring and reporting options. The main area displays 'Service Status Details For All Hosts'. It lists several hosts (google, mailserver, nra, pegaso, windows) with their corresponding services (ping, http, ssh, etc.) and their current status (OK, WARNING, PENDING, UNKNOWN, CRITICAL). Each row includes columns for 'Host', 'Service', 'Status', 'Last Check', 'Duration', 'Attempt', and 'Status Information'. Below this, there's a note indicating 14 matching service entries displayed.

Figura 4. Configuración Personalizada de Nagios

The screenshot shows the Nagios configuration interface. The sidebar is identical to Figure 4. The main area displays 'Host Status Details For All Host Groups'. It lists several host groups (google, mailserver, nra, pegaso, windows) with their current status (UP or DOWN). Each row includes columns for 'Host Group', 'Status', 'Last Check', 'Duration', and 'Status Information'. Below this, there's a note indicating 5 matching host entries displayed.

Figura 5. Estado de Dispositivos



Perl

Perl es un sistema de programación en modo texto que se utiliza para gestión de tareas del sistema.

```
$ sudo apt-get install libconfig-inifiles-perl
```

SNMP

Centreon necesita el protocolo SNMP, así como el módulo libnet-snmp-perl para recoger o actualizar información en una máquina remota utilizando SNMP (Simple Network Management Protocol o Protocolo Simple de Administrador de Red) para su operativa. Lo instalaremos de la siguiente manera:

```
$ sudo apt-get install snmp snmpd libnet-snmp-perl
```

Y añadiremos la siguiente línea en el archivo de configuración snmpd.conf:

```
$ sudo vi /etc/snmp/snmpd.conf
com2sec  readonly  default
public
```

Reiniciaremos el demonio snmpd:

```
$ sudo vi /etc/init.d/snmpd restart
```

Y comprobaremos que funciona correctamente:

```
$ snmpwalk -v1 -c public localhost
SNMPv2-MIB::sysDescr.0 = STRING:
Linux jose-laptop 2.6.22-14-generic
#1 SMP Tue Dec 18 08:02:57 UTC 2007
i686
SNMPv2-MIB::sysObjectID.0 = OID:
NET-SNMP-MIB::netSnmpAgentOIDs.10
...
End of MIB
```

Módulos Pear Adicionales

Pear es una interfaz para descargar e instalar componentes php. De alguna manera realiza las funciones de apt, pero en vez de descargar e instalar paquetes .deb, descarga e instala aquellos perl necesarios para nuestro sistema.

Ejecutaremos los siguientes comandos para actualizar pear y para instalar los componentes necesarios respectivamente:

```
# pear upgrade pear
# pear install -o -f --alldeps DB_
DataObject \
```

```
DB_DataObject_FormBuilder MDB2
Numbers_Roman \
Numbers_Words HTML_Common HTML_
QuickForm \
HTML_QuickForm_advmultiselect HTML_
Table Auth_SASL \
HTTP Image_Canvas Image_Color Image_
Graph Image_GraphViz \
Net_Traceroute Net_Ping Validate
XML_RPC SOAP
```

Instalación de Centreon

La instalación de Centreon no es lo sencilla que debería y puede crear más de un dolor de cabeza al lector si no se siguen los pasos correctamente.

Es importante aclarar, que en este artículo se describe la instalación para Nagios 2.

Antes de empezar, descargaremos la última versión de Centreon. 1.4.2.3 A la hora de escribir este artículo:

```
$ wget http://download.oreon-project.org/centreon/centreon-1.4.2.3.tar.gz
```

Lo descomprimimos y cambiamos al directorio de instalación:

```
$ tar xvzf centreon-1.4.2.3.tar.gz
$ cd centreon-1.4.2.3
```

No podremos instalar Centreon sin arreglar un problema en su script `install.sh`, el cual busca el binario nagios en vez de nagios2. Para ello lo editaremos, iremos a la línea 293 del archivo y cambiaremos `nagios` por `nagios2`. (Ver texto en negrita).

3. Environment Configuration



In order for your Oreon installation to function properly, please complete the following fields.

Environment Configurations

Nagios user	<input type="text" value="nagios"/>
Nagios group	<input type="text" value="nagios"/>
Apache User	<input type="text" value="www-data"/>
Apache Group	<input type="text" value="www-data"/>
Nagios Version	<input type="text" value="2.x"/>
Nagios configuration directory	<input type="text" value="/etc/nagios2/"/>
Nagios plugins	<input type="text" value="/usr/lib/nagios/plugins/"/>
RRDTool binary	<input type="text" value="/usr/bin/rrdtool"/>

[Back](#) | [Next](#)

Figura 7. Parámetros de Entorno de Centreon

11. Creating Database



Component	Status
Database : Connection	OK
Database 'centreon' : Creation	OK
Database 'cds' : Creation	OK
Database 'centreon' : Users Management	OK
Database 'centreon' : Schema Creation	OK
Database 'cds' : Schema Creation	OK
Database 'centreon' : Macros Creation	OK
Database 'centreon' : Insert Basic Configuration	OK
Database 'centreon' : Insert Commands	OK
Database 'centreon' : Topology Insertion	OK
Database 'centreon' : Centreon User Creation	OK
Database 'centreon' : Customization	OK

[Back](#) | [Next](#)

Figura 8. Creación de Base de Datos en Centreon



seguridad

Monitorización de sistemas

```
while [ ! -x "${temp}/nagios2" ]; do
    echo_passed "Cannot find
${temp}/nagios2"
"CRITICAL"
```

Comenzaremos la instalación ejecutando el comando tal y como se indica a continuación:

```
$ sudo bash install.sh
```

El programa de instalación nos da la bienvenida y nos advierte de la pérdida de datos si tuviéramos una instalación previa. Como es instalación nueva, pulsamos y (yes).

En el Listado 10. Instalación de Centreon se muestran en negrita los cambios a realizar durante el proceso de instalación.

Configuración Inicial

Será en este momento cuando empezaremos a configurar Centreon. Abriremos nuestro navegador web y teclearemos en la barra de direcciones la URL `http://localhost/centreon/install/setup.php`. Nos aparecerá la pantalla de bienvenida de Centreon. Pulsaremos sobre el botón *Start* y nos aparecerá la pantalla de Licencia, donde marcaremos la opción *I Accept* si estamos de acuerdo y pulsaremos el botón *Next* para continuar.

La siguiente pantalla nos permitirá configurar los parámetros de entorno, donde no tendremos que cambiar nada, aunque revisaremos los valores que nos ofrece Centreon, ver Figura 7.

Pulsamos *Next*, y el sistema chequeará la configuración. Pulsando de nuevo *Next*,

chequeará los componentes PHP instalados.

Llegamos a la configuración de la base de datos, donde indicaremos los siguientes parámetros (ver Tabla 1).

Pulsamos *Next*. El asistente verificará la versión de MySQL, donde pulsando de nuevo *Next*, llegaremos a la pantalla de configuración de la interfaz de Centreon. En esta pantalla se nos pide los siguientes datos:

Al pulsar *Next*, nos aparecerá una pantalla para activar y configurar LDAP. Le diremos que *no* (por defecto) y volvemos a pulsar *Next*.

Nos aparecerá una pantalla indicando el estado de los archivos de configuración de Centreon. Pulsamos *Next*.

Centreon creará la base de datos con los parámetros dados durante el asistente y mostrará una pantalla de estado, ver Figura 8.

Pulsando *Next* nos aparecerá la última pantalla del proceso de configuración. Pulsamos el botón *Click here to complete your install* para finalizar la configuración. En este momento, se nos mostrará la pantalla de acceso.

Antes de escribir nuestro usuario (en nuestro ejemplo *admin*) y nuestra contraseña, reiniciaremos Nagios con el comando

```
$ sudo /etc/init.d/nagios2 restart
```

Una vez que accedamos a Centreon, se nos mostrará la pantalla de inicio, donde seleccionando la opción Monitoring – Service se nos mostrarán los servicios previamente configurados en Nagios, ver Figura 9.

Ventajas e Inconvenientes de Centreon

Aunque ciertamente Centreon mejora de una forma notable la interfaz web de Nagios, como podéis ver, su instalación es realmente compleja y no menos su configuración, la cual puede acarrear más de un dolor de cabeza. He echado de menos buena documentación, tanto para instalar como para configurar, pero como veis, nada que no pueda resolverse de una forma más o menos fácil.

La pregunta sobre Centreon es, qué pasará en un futuro próximo, su soporte con la futura base de datos MySQL de Nagios3 y su adaptación con esa nueva versión, aunque no tengo ninguna duda de que lo hará sin ningún problema.

Conclusiones

Hace sólo unos años era imposible imaginar que el Software Libre pudiera conseguir controlar y manejar de una forma casi automática cualquier red de cualquier tamaño junto con sus aplicaciones. Hoy podemos decir que tanto Nagios, Zabbix o Zenoss han conseguido que con un coste mínimo podemos estar tranquilos sabiendo que nuestros equipos funcionarán y serán controlados por estos sistemas proactivos de Monitorización de la forma más óptima.

Estos tres sistemas son, sin duda, una dura competencia para el software comercial, dando a cualquier usuario o empresa de un sistema GNU/Linux, cualquier característica que pueda requerir el más exigente: Datos y Alarmas, Rendimiento y Flexibilidad.

The screenshot shows the Centreon web interface. At the top, there's a navigation bar with links like 'Home', 'Monitoring', 'Reporting', 'Centreon', 'Views', 'ID Cards', 'Options', and 'Configuration'. Below the navigation is a search bar and a user dropdown. The main content area has tabs for 'Services' (selected), 'Hosts', and 'Event Log'. Under 'Services', there's a table with columns: Hosts, Services, Infos, Status, Last Check, Duration, and Tries. The table lists various services like google, ping, and http, each with a status icon (green for OK, yellow for warning, red for critical). The 'Status Information' column provides more details for each service. At the bottom of the page, there's a footer with copyright information and a link to 'AdBlock'.

Figura 9. Estado de Servicios en Centreon



Sobre el Autor

José Ángel Muñoz Martínez es Técnico Superior de Informática de Sistemas. Nacido en Madrid, actualmente trabaja en el sector de la RFID (Identificación por Radio Frecuencia) en Athelia Solutions, empresa líder en este sector. A parte de la lectura y los viajes, dentro de la informática sus intereses son, el software libre, Linux y todo lo relacionado con las comunicaciones.

Su blog personal está en <http://linux-neobook.blogspot.com/> y su correo, josea.munoz@gmail.com