

NRDP - Nagios Remote Data Processor.

Introducción.

Históricamente el método que teníamos de enviar comandos externos y chequeos de hosts / servicios a Nagios era a través del "addon" NSCA. La función de poder enviar comandos a Nagios surge básicamente para cubrir una serie de necesidades:

- Chequeos pasivos: es el host remoto el que manda el resultado de los chequeos a Nagios. Este tiene definidos los servicios como pasivos y un tiempo de espera entre recepción de resultados para alertar en caso de que no lleguen. El uso de chequeos pasivos es útil entre otros en los siguientes casos:
 - Hosts que están detrás de Firewalls y no están accesibles desde el servidor de Nagios pero si al revés.
 - Configuraciones de monitorización distribuidas, donde otros servicios / aplicaciones se encargan de realizar los chequeos y enviárselos como pasivos a Nagios.
- Control remoto de funcionalidades / comportamientos de Nagios. Mediante [comandos externos](#) podemos alterar varios aspectos y funcionalidades de Nagios además de alterar configuraciones de ciertos objetos. Puedes sorprenderte de de la lista de [comandos externos disponibles](#).

NRDP se diseñó como una evolución de NSCA para solucionar una serie de problemas que este tenía. Las ventajas que proporciona NRDP sobre NSCA son:

- Usa puertos estándar y protocolos web. Esto hace que simplifique la configuración de Firewalls y el desarrollo de clientes.
- Usa el servidor Apache para proporcionar autenticación y encriptación SSL (opcional).
- Soporta la salida multi línea que implementan numerosos plugins y que ya soportaba la versión 3 de Nagios.
- Escribe la salida de los chequeos al directorio spool de Nagios (no usa el fichero de comandos externos). Esto mejora el rendimiento.

Instalación de NRDP Server.

[NRDP Server](#) es básicamente una serie de scripts para su ejecución vía web. Descomprimos el ZIP en el directorio donde queremos ubicarlo. Nos creará un directorio /usr/local/nrdp. Le asignamos permisos al directorio para nuestro usuario y grupo de Nagios.

```
unzip nrdp.zip -d /usr/local/  
chown -R nagios.nagios /usr/local/nrdp
```

Editamos el fichero de configuración “/usr/local/nrdp/server/config.inc.php” y modificamos / revisamos:

- Le añadimos al menos un texto / token que no sea demasiado sencillo si estamos en un sistema en producción
- Configuramos la ruta de nuestro fichero `command_file` de Nagios.
- Configuramos la ruta de nuestro directorio “checkresults” de Nagios.
- Creamos un directorio al que tengan acceso de lectura/escritura tanto nuestro usuario de Apache como el de Nagios. Un ejemplo podría ser:

```
mkdir /var/log/nagios/spool/tmp
chown apache.nagios /var/log/nagios/spool/tmp
chmod 660 /var/log/nagios/spool/tmp
```

Un ejemplo de fichero configurado para un sistema concreto sería:

```
$cfg['authorized_tokens'] = array(
    "123456abcde",
);
$cfg["nagios_command_group"]="nagios";
// full path to Nagios external command file
$cfg["command_file"]="var/spool/nagios/cmd/nagios.cmd";
// full path to check results spool directory
$cfg["check_results_dir"]="var/log/nagios/spool/checkresults";
// full path to directory where temp scratch files can be written
// NOTE: the Apache user need to be able create files here, and the Nagios user needs to read/delete those same files, so
the /tmp system directory won't work (it has a sticky bit on it)
$cfg["tmp_dir"]="var/log/nagios/spool/tmp";
```

Copiamos el fichero de configuración de Apache en nuestro directorio correspondiente de ficheros de configuración de este y recargamos la configuración.

```
cp /usr/local/nrdp/nrdp.conf /etc/httpd/conf.d
service httpd reload
```

Probamos a acceder al servidor nrpd y metemos el token para probar
http://nuestro_servidor/nrdp

Submit Nagios Command:
Token:
Command:

Submit Check Data
Token:
Check Data:

```
<?xml version='1.0'?>
<checkresults>
  <checkresult type='host'>
    <hostname>somehost</hostname>
    <state>0</state>
    <output>Everything looks okay!|perfddata</output>
  </checkresult>
  <checkresult type='service'>
    <hostname>somehost</hostname>
    <servicename>someService</servicename>
    <state>1</state>
    <output>WARNING: Danger Will Robinson!|perfddata</output>
  </checkresult>
</checkresults>
```

Si nos aparece un error del tipo “BAD CHECK RESULTS DIR ” o algo similar relacionado con directorios / ficheros es que hemos configurado algo mal relativo a las rutas en el fichero de configuración “config.inc.php”.

Tanto en “Submit Command” como en “Submit Check Data” nos debe dar un OK.

```
▼<result>
  <status>0</status>
  <message>OK</message>
  ▼<meta>
    <output>2 checks processed.</output>
  </meta>
</result>
```

En el fichero log de Nagios podemos ver la evolución de los comandos que enviamos. Si llegan correctamente, si los asocia con un Host / Servicio existente,...

Creación de Host / Servicios de prueba.

Antes de ponernos a probar los clientes vamos a definir un par de Host y unos servicios pasivos para probar nuestro servidor / clientes NRDP. Si quieres saber más sobre [cheques pasivos...](#) Definiremos un servicio adicional de PING para chequear el servidor (no necesario) y un chequeo “falso” que se ejecutará cuando pase un tiempo X sin recibir resultados. De esta forma nos enteraremos si no estamos recibiendo resultados pasivos desde el Host remoto.

```
define host {
    use                windows-server
```

```

    host_name      winsrv-nrdp
    alias          winsrv-nrdp-test
    address        192.168.1.33
}

define host{
    use            linux-server
    host_name      centos-nrdp
    alias          centos-nrdp-test
    address        192.168.1.43
}

define service{
    use            generic-service
    host_name      winsrv-nrdp,centos-nrdp
    service_description    PING
    check_command  check_ping!100.0,20%!500.0,60%
    passive_checks_enabled  1
}

define service{
    use            generic-service
    host_name      winsrv-nrdp,centos-nrdp
    service_description    CHECK_PASIVO
    check_command  sin_resultados
    passive_checks_enabled  1
    active_checks_enabled  0
    check_freshness        1
    freshness_threshold    600 ; Umbral de 10 minutos, si no recibe informacion ejecutara el
check_command
}

define command{
    command_name sin_resultados
    command_line /usr/lib64/nagios/plugins/check_dummy 2 "CRITICAL: No se han recibido resultados en
tiempo"
}

```

Cientes NRDP

Cliente php para Linux

El paquete de NRDP incluye un cliente básico para enviar comandos remotamente. Es un script PHP que está en el directorio “./clients/send_nrdp.php”. También deberíamos poder usarlo en Windows. Requiere, claro, que tengamos intérprete de PHP en el servidor. Esto puede ser un problema por eso existen otras alternativas de terceros que veremos luego.

Al ejecutar el script PHP en la versión usada me encontré con un problema cuanto menos extraño. Daba un error tal que: “Exception: Zend Extension ./send_nrdp.php does not exist”. Es posible que el fichero este editado desde Windows y tenga caracteres raros porque la solución consistió en hacerle un “dos2unix” al fichero.

```
dos2unix send_nrdp.php
```

La ayuda como siempre con: “send_nrdp.php --help”

Un comando de ejemplo para nuestros servicios:

```
./send_nrdp.php --url=http://192.168.1.43/nrdp/ --token=123456abcde --host=centos-nrdp --service=CHECK_PASIVO --state=0 --output="Todo parece ir estupendo"
```

Este es el aspecto que tendrían nuestros servicios si no reciben resultados en el tiempo prefijado.

Host	Service	Status	Status Information
centos-nrdp	CHECK_PASIVO	? CRITICAL	CRITICAL: CRITICAL: No se han recibido resultados en tiempo
	PING	OK	PING OK - Packet loss = 0%, RTA = 0.05 ms
winsrv-nrdp	CHECK_PASIVO	? CRITICAL	CRITICAL: CRITICAL: No se han recibido resultados en tiempo
	PING	OK	PING OK - Packet loss = 0%, RTA = 0.62 ms

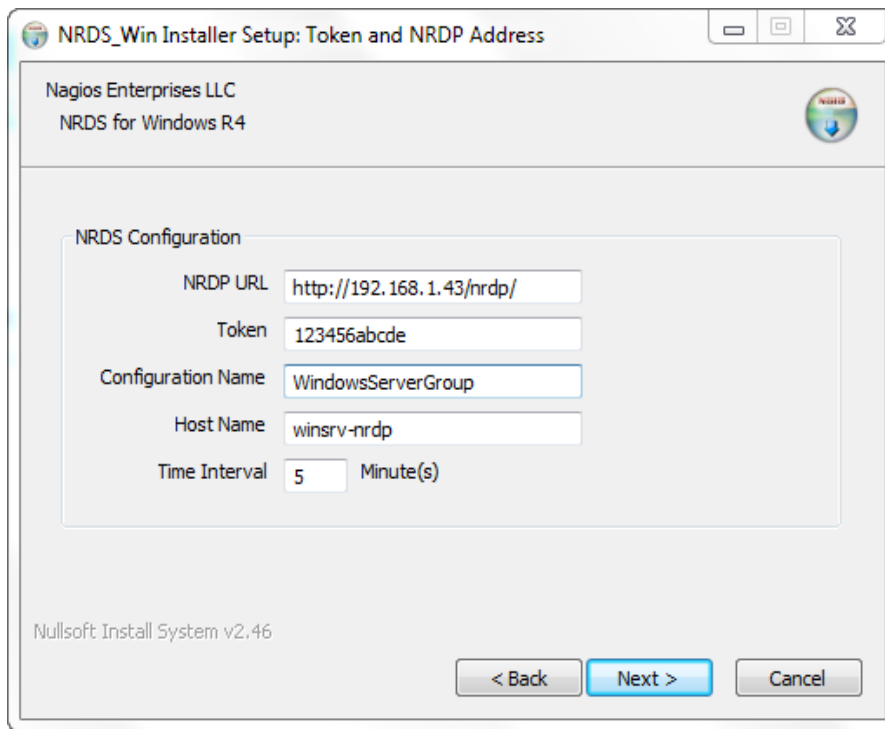
Y este cuando reciban nuestros comandos por nrpe.

Host	Service	Status	Lá	Status Information
centos-nrdp	CHECK_PASIVO	? WARNING	0	Hay algo como que va raro...
	PING	OK	01	PING OK - Packet loss = 0%, RTA = 0.04 ms
winsrv-nrdp	CHECK_PASIVO	? OK	0	Todo parece ir estupendo
	PING	OK	01	PING OK - Packet loss = 0%, RTA = 0.82 ms

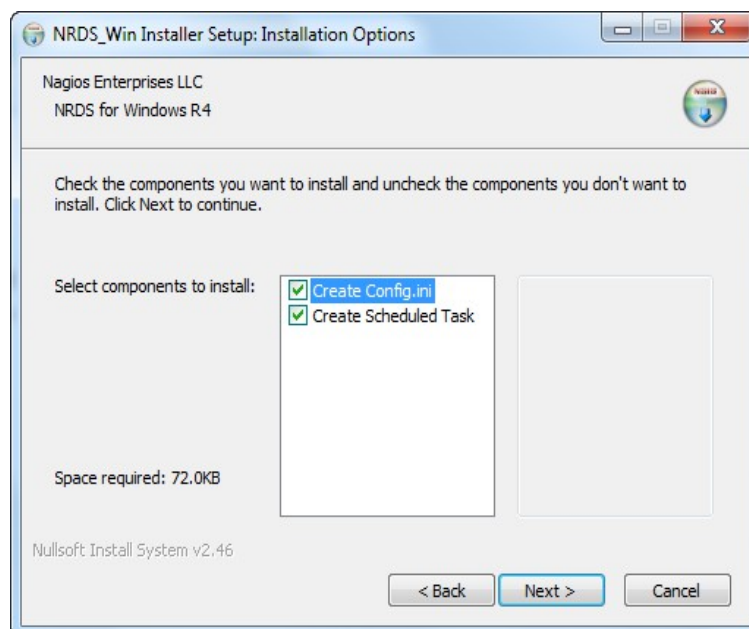
Cliente para Windows.

Nagios Enterprise dispone de un cliente para Windows denominado NRDS (Nagios Remote Data Sender for Windows) que podemos usar tanto para Nagios XI como para Nagios Core (aunque para este último perdemos las características de automatización que si tenemos con Nagios XI).

La instalación es muy sencilla, pide una serie de parámetros que además posteriormente podemos modificar en su correspondiente fichero INI.



En nuestro caso que estamos probándolo en Nagios Core perdemos como dijimos las características de automatización así que el valor de “Configuration Name” no nos valdrá de mucho (asocia en Nagios XI a un grupo de distribución de configuración y Plugins). Tampoco recibiremos de este los Plugins disponibles de forma automática por lo que tendremos que copiarlos a mano. Podemos bajar algunos plugins para Windows de [este enlace](#).



En este paso del Asistente dejamos las opciones por defecto de crear fichero ini y tareas programadas (verificar que ha creado la tarea winsrv-ndrp y está programada cada X minutos) para que envié los resultados de los checks al servidor NRDP en los intervalos de tiempo especificados. Posteriormente podemos editar y modificar dichos parámetros en C:\Program Files\Nagios\NRDS_Win\config.ini

Podemos copiar los plugins si que se pueden distribuir colocándolos en la carpeta “Plugins”.

La configuración de los chequeos a realizar se configura también en el fichero “config.ini”. Vemos un sencillo ejemplo:

```
[Services]
#<check_name> = <path location\file name> <check arguments>
CHECK_PASIVO = .\plugins\cpuload_nrpe_nt.exe 70 90
```

winsrv-ndrp	CHECK_PASIVO	?	OK	NOW: Mean:30.761906% Variance: 20.738684% CUMULATIVE: Mean:25.105263% Variance: 14.346927%
	PING		OK	PING OK - Packet loss = 0%, RTA = 0.65 ms

Por supuesto el servicio quedaría mejor llamándole algo así como “CPU Load” :-)

Podremos usar en teoría cualquier plugin estándar para windows que devuelva valor de retorno y cadena de salida. También scripts vbs y powershell. Lo interesante de NRDS es que nos vale cualquier plugin estándar ya que el software se encarga de gestionar la salida de este y mandarla al servidor NRPE.

Otros clientes.

Una sencilla búsqueda en [Nagios Exchange](#) por “NRDP” nos mostrará numerosos clientes para enviar resultados pasivos a NRDP. Los hay en Python, Perl, Powershell, bash,... Solo tenemos que probarlos y quedarnos con el que nos convenga.

Un ejemplo en Python es “[send_nrpd Python NRDP Client](#)” muy sencillo y eficiente. Lo bajamos a nuestra máquina y marcamos ejecutable. Probamos un par de chequeos.

```
./send_nrpd.py -u http://192.168.1.43/nrdp/ -t 123456abcde -H winsrv-ndrp -s CHECK_PASIVO -S 0 -o "Todo parece ir estupendo" -c 1

./send_nrpd.py -u http://192.168.1.43/nrdp/ -t 123456abcde -H centos-ndrp -s CHECK_PASIVO -S 1 -o "Hay algo como que va raro..." -c 1
```

El resultado:

Host	Service	Status	Lá	Status Information
centos-nrdp	CHECK_PASIVO	?	0	Hay algo como que va raro...
	PING	OK	01:	PING OK - Packet loss = 0%, RTA = 0.04 ms
winsrv-nrdp	CHECK_PASIVO	?	0	Todo parece ir estupendo
	PING	OK	01:	PING OK - Packet loss = 0%, RTA = 0.82 ms

Y ahora te preguntará... ¿Y como uso yo plugins estándar para mandar el resultado por NRPE? Buena pregunta. Con el cliente para Windows NRDS hemos visto que basta con usar plugins en formato estándar y configurarlos en un fichero ini. Parece que no hay similar a mano para Linux por lo que tendremos que usar un script previo para realizarlo. Con una búsqueda por “nrdp_wrapper” localizarás un script que te permite hacerlo pero tendrás que llamar con este a cada plugin que vayas a usar para mandar resultados a NRDP. Puedes añadirseles posteriormente al cron con el intervalo deseado y listo. El paquete de NRDP para Nagios XI si que se baja, al instalarlo, entre configuraciones y plugins para diferentes S.O., un script en perl para realizar esto mismo.

Enlaces de Interés.

[NRDP Overview.](#)

[Presentación de NRDS](#) en diapositivas.

Tutorial en video de [Nagios Remote Data Sender \(NRDS\)](#) para Nagios XI.

[Documentación de NRDS.](#)