

Installation et configuration de Nagios pour débutants

Apprendre par l'exemple

Par djibril 

Date de publication : 26 janvier 2011

Dernière mise à jour : 10 juin 2017

TOUT PUBLIC

Vous souhaitez vous lancer dans l'installation de Nagios, mais vous êtes perdu dans la documentation officielle assez volumineuse et les différents tutoriels sur la toile pas toujours compréhensibles ?

Cet article va vous guider à réaliser une installation simplifiée de Nagios et une configuration pour superviser une machine Linux, une machine Windows et un switch.

Commentez

En complément sur Developpez.com

- [Apprendre à surveiller certains services avec Nagios](#)
- [Concevoir facilement un plugin Nagios en Perl](#)

I - Introduction.....	3
I-A - Exemple de réseau à superviser.....	3
I-B - Les prérequis.....	3
I-B-1 - Bibliothèques nécessaires.....	3
II - Installation de Nagios.....	4
II-A - Création des comptes et groupes nécessaires.....	4
II-B - Téléchargement de Nagios et de ses plug-ins.....	4
II-C - Compilation et installation de Nagios.....	5
II-D - Personnalisation de la configuration de Nagios.....	7
II-E - Configuration de l'interface Web.....	8
II-F - Compilation et installation des plug-ins Nagios.....	8
II-G - Démarrage de Nagios.....	8
III - Configuration de Nagios.....	15
III-A - Surveillance locale.....	16
III-B - Surveillance à distance.....	17
III-B-1 - Machines à surveiller sous Windows.....	17
III-B-1-a - Agent NSClient++ jusqu'en avril 2012.....	18
III-B-1-b - Agent NSCP à partir de mai 2012.....	19
III-B-1-c - Configuration de Nagios.....	21
III-B-2 - Machines à surveiller sous Linux.....	23
III-B-2-a - Installation de NRPE et des plug-ins Nagios sur le serveur Linux distant.....	23
III-B-2-b - Installation de NRPE sur le serveur Nagios.....	30
III-B-3 - Switch à surveiller.....	33
IV - Mise à jour de Nagios 2.x/3.x vers 3.4.1, 4.x vers 4.x.....	35
V - Divers.....	37
V-A - Installation openssl, openssl-devel.....	37
V-B - Erreurs.....	37
V-B-1 - Could not complete SSL handshake.....	37
V-B-2 - Page URL Could not complete SSL handshake.....	38
V-B-3 - Page supervision not found.....	39
V-B-4 - Could not connect to xxx.xxx.xxx.xxx : Connection reset by peer.....	39
VI - Liens.....	40
VII - Conclusion.....	40
VIII - Remerciements.....	41

I - Introduction

Le but de ce tutoriel est d'installer un serveur de supervision pour détecter les pannes ou incohérences et récupérer des informations sur des serveurs Windows, Linux ou switch et autres éléments d'un réseau. L'outil de supervision que nous utiliserons sur notre parc informatique est le logiciel open source Nagios 4.3.2 et le serveur sur lequel l'installation sera faite aura un système d'exploitation Linux Debian 8.8.

Pour la rédaction, je me suis basé sur la documentation officielle de Nagios, d'autres cours trouvés sur la toile, ainsi qu'un peu d'huile de coude !! Vous aurez en fin d'article un résumé de ces différents liens.

I-A - Exemple de réseau à superviser

Le réseau que nous allons superviser est de type réseau LAN et contient trois machines et un switch, il dispose également d'un serveur DNS.

- Un serveur Linux nommé « supervision » sur lequel sera installé Nagios pour superviser notre réseau. Il aura pour IP 192.168.1.12. Il est sous Debian 8.8.

```
## uname -a
Linux supervision 3.16.0-4-686-pae #1 SMP Debian 3.16.43-2 (2017-04-30) i686 GNU/Linux
# cat /etc/debian_version
8.8
```

- Un serveur Linux « servprod » qui sera supervisé (sous Debian). Il aura pour IP 192.168.1.10.
- Un serveur Windows « winprod » qui sera supervisé. Il aura pour IP 192.168.1.11.
- Un switch 3com4300 nommé « switch » que l'on supervisera également. Il aura pour IP 192.168.1.1.

Bien que les serveurs soient sous Debian, dans ce tutoriel, vous trouverez quelques commandes pour une distribution Suse ou autres.



En ce qui concerne la machine Windows, elle peut être sous XP, Vista, Seven, Windows 2008...

I-B - Les prérequis

I-B-1 - Bibliothèques nécessaires

Nous avons besoin d'installer quelques bibliothèques utiles au bon fonctionnement de Nagios, mais surtout à son installation, compilation...

Nous devons installer apache2, php et gd, utiles pour la future interface de Nagios. Il nous faut aussi make et gcc pour les compilations et snmp pour superviser les routeurs, switch...

Voici un exemple de commandes à lancer pour installer nos prérequis :

Sous Suse, utilisez YaST et recherchez apache2, apache2-mod-php5, php5, gcc, Net-SNMP.

Sous Debian ou Ubuntu, vous devriez pouvoir installer ces bibliothèques avec apt-get, ou yum sous RedHat.

Debian et Ubuntu sous root

```
# apt-get update
# apt-get upgrade
# apt-get install unzip apache2 libapache2-mod-php5 php5-gd php5 make gcc build-essential wget
libgd-gd2-perl libgd2-dev libgd2-xpm libgd2-xpm-dev libnet-snmp-perl libssl-dev snmp daemon
```

Debian et Ubuntu sous root

```
# apt-get install autoconf bc gawk dc libc6 make gcc libmcrypto-dev build-essential wget libssl-dev snmpd snmp scli xinetd snmp libnet-snmp-perl gettext
```

Redhat sous root

```
# yum update
# yum install -y make wget httpd php gcc glibc glibc-common gd gd-devel gd2 gd2-devel net-snmp snmp
# yum install openssl openssl-devel
# yum install xinetd
```

II - Installation de Nagios

Installer Nagios revient à créer un utilisateur et télécharger et installer deux archives (Nagios et ses plug-ins). Les installations seront faites depuis les sources officielles et non des paquets tout prêts.

II-A - Création des comptes et groupes nécessaires

Création d'un utilisateur « nagios » ayant pour mot de passe « nagios » et appartenant à un groupe nommé « nagios ».

Création du groupe nagios

```
supervision:~ # groupadd nagios
```

Création du compte utilisateur « nagios » que nous allons mettre dans le groupe « nagios »

Sous OpenSuse

```
supervision:~ # useradd -m -G nagios nagios
```

Sous Debian ou RedHat

```
supervision:~ # useradd -m -g nagios nagios
```

Attribution d'un mot de passe à l'utilisateur « nagios ». Ce mot de passe est « nagios ».

```
supervision:~ # passwd nagios
```

Création d'un groupe « nagcmd » permettant l'exécution des commandes externes à travers l'interface Web. Rajout des utilisateurs Nagios et Apache à l'intérieur du groupe « nagcmd ».

Sous OpenSuse

```
supervision:~ # groupadd nagcmd
supervision:~ # usermod -G nagcmd nagios
supervision:~ # usermod -G nagcmd wwwrun
```

Sous Debian

```
supervision:~ # groupadd nagcmd
supervision:~ # usermod -g nagcmd nagios
supervision:~ # usermod -g nagcmd www-data
```

N.B. Sous OpenSuse, le compte utilisateur d'Apache est wwwrun et son groupe est www. Maintenant que les comptes ont été créés, nous allons procéder aux téléchargements et l'installation des bibliothèques Nagios.

II-B - Téléchargement de Nagios et de ses plug-ins

Nous avons besoin de Nagios et de ses plug-ins. Nous les téléchargerons sur le [site de Nagios](#).
Création d'un répertoire /nagios/download dans lequel nous déposerons les archives à installer.

Nagios et ses plug-ins

```

root@supervision:~# mkdir -p /nagios/download
root@supervision:~# cd /nagios/download
root@supervision:/nagios/download# wget https://sourceforge.net/projects/nagios/files/nagios-4.x/nagios-4.3.2/nagios-4.3.2.tar.gz
root@supervision:/nagios/download# wget http://www.nagios-plugins.org/download/nagios-plugins-2.2.1.tar.gz
root@supervision:/nagios/download# ls -alh
total 4,0M
drwxr-xr-x 2 root root 4,0K oct. 15 11:47 .
drwxr-xr-x 3 root root 4,0K oct. 15 11:45 ..
-rw-r--r-- 1 root root 1,8M août 12 2014 nagios-4.3.2.tar.gz
-rw-r--r-- 1 root root 2,4M oct. 2 17:27 nagios-plugins-2.2.1.tar.gz
    
```

À ce jour (10/06/2017), la version la plus récente de Nagios est la 4.3.2 et celle du « plug-in Nagios » est la 2.2.1.

N.B. Nagios vous donne la possibilité de créer vos propres plug-ins via différents langages de programmation, notamment Perl. Pour en savoir plus, consultez cet article : [Concevoir facilement un plug-in Nagios en Perl](#).

II-C - Compilation et installation de Nagios

Il faut extraire le code source de l'archive de Nagios dans un premier temps.

```

supervision: # cd /nagios/download/
root@supervision:/nagios/download# tar -xzf nagios-4.3.2.tar.gz
    
```

Puis exécuter le script de configuration en lui précisant le nom du groupe créé précédemment.

```

supervision:/nagios/download # cd nagios-4.3.2/
supervision:/nagios/download/nagios # ./configure --with-command-group=nagcmd
    
```

Compilons le code source de Nagios.

```

supervision:/nagios/download/nagios-4.3.2 # make all
    
```



Si « make » n'est pas trouvé, installez-le et recommencez.

Vous ne devez pas avoir de message d'erreurs, à la rigueur, des warnings.

Installons les binaires.

```

supervision:/nagios/download/nagios-4.3.2 # make install
    
```

Installons les scripts de démarrage.

```

root@supervision:/nagios/download/nagios-4.3.2# make install-init
/usr/bin/install -c -m 755 -d -o root -g root /etc/init.d
/usr/bin/install -c -m 755 -o root -g root daemon-init /etc/init.d/nagios

*** Init script installed ***

sous Debian 7.x

root@supervision:/nagios/download/nagios-4.3.2# update-rc.d nagios defaults

ou sous Debian 8.x
root@supervision:/nagios/download/nagios-4.3.2# update-rc.d nagios defaults
    
```

Le fichier « /etc/init.d/nagios » est créé.

Installons les fichiers de configuration.

Si vous faites une mise à jour, il est important de faire une sauvegarde des fichiers suivants :



- /usr/local/nagios/etc/nagios.cfg ;
- /usr/local/nagios/etc/nrpe.cfg ;
- tous les fichiers du répertoire /usr/local/nagios/etc/objects où sont configurés vos commandes nagios.

L'écrasement ou la perte de ces fichiers vous donnera des sueurs froides !

Les fichiers seront automatiquement installés dans le répertoire /usr/local/nagios/etc.

```
root@supervision:/nagios/download/nagios-4.3.2# make install-config
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/etc
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/etc/objects
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/nagios.cfg /usr/local/nagios/etc/nagios.cfg
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/cgi.cfg /usr/local/nagios/etc/cgi.cfg
/usr/bin/install -c -b -m 660 -o nagios -g nagios sample-config/resource.cfg /usr/local/nagios/etc/resource.cfg
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/template-object/templates.cfg /usr/local/nagios/etc/objects/templates.cfg
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/template-object/commands.cfg /usr/local/nagios/etc/objects/commands.cfg
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/template-object/contacts.cfg /usr/local/nagios/etc/objects/contacts.cfg
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/template-object/timeperiods.cfg /usr/local/nagios/etc/objects/timeperiods.cfg
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/template-object/localhost.cfg /usr/local/nagios/etc/objects/localhost.cfg
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/template-object/windows.cfg /usr/local/nagios/etc/objects/windows.cfg
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/template-object/printer.cfg /usr/local/nagios/etc/objects/printer.cfg
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/template-object/switch.cfg /usr/local/nagios/etc/objects/switch.cfg

*** Config files installed ***

Remember, these are *SAMPLE* config files. You'll need to read the documentation for more information on how to actually define services, hosts, etc. to fit your particular needs.
```

Installons et configurons les permissions.

```
root@supervision:/nagios/download/nagios-4.3.2# make install-commandmode
/usr/bin/install -c -m 775 -o nagios -g nagcmd -d /usr/local/nagios/var/rw
chmod g+s /usr/local/nagios/var/rw

*** External command directory configured ***
```



À ce stade, il ne faut surtout pas démarrer Nagios.

II-D - Personnalisation de la configuration de Nagios

Les fichiers de configuration se trouvent dans le répertoire `/usr/local/nagios/etc`. Éditez le fichier `contacts.cfg` pour y mettre les informations de l'administrateur de Nagios, notamment l'adresse électronique où les alertes seront envoyées.

Fichiers `/usr/local/nagios/etc/objects/contacts.cfg`

```

root@supervision:/nagios/download/nagios-4.3.2# vi /usr/local/nagios/etc/objects/contacts.cfg

#####
# CONTACTS.CFG - SAMPLE CONTACT/CONTACTGROUP DEFINITIONS
#
# Last Modified: 05-31-2007
#
# NOTES: This config file provides you with some example contact and contact
#        group definitions that you can reference in host and service
#        definitions.
#
#        You don't need to keep these definitions in a separate file from your
#        other object definitions. This has been done just to make things
#        easier to understand.
#
#####

#####
#####
#
# CONTACTS
#
#####
#####

# Just one contact defined by default - the Nagios admin (that's you)
# This contact definition inherits a lot of default values from the 'generic-contact'
# template which is defined elsewhere.

define contact{
    contact_name    nagiosadmin                ; Short name of user
    use              generic-contact           ; Inherit default values from generic-contact
    template (defined above)
    alias           Administrateur Nagios      ; Full name of user
    email           toto@tutu.fr ; Adresse Email pour les notifications
}

#####
#####
#
# CONTACT GROUPS
#
#####
#####

# We only have one contact in this simple configuration file, so there is
# no need to create more than one contact group.

define contactgroup{
    contactgroup_name    admins
    alias                Administrateurs Nagios
    members              nagiosadmin
}
    
```

Le renseignement de l'adresse mail ne suffit pas pour que les envois de mails se fassent par Nagios. Il faut que le serveur ait un serveur de messagerie activé (sendmail, postfix...). Mais ce n'est pas le but de ce tutoriel.

II-E - Configuration de l'interface Web

Installation du fichier de configuration de Nagios dans le répertoire conf.d d'Apache.

```
root@supervision:/nagios/download/nagios-4.3.2# make install-webconf
/usr/bin/install -c -m 644 sample-config/httpd.conf /etc/apache2/conf.d/nagios.conf
if [ 0 -eq 1 ]; then \
    ln -s /etc/apache2/conf.d/nagios.conf /etc/apache2/sites-enabled/nagios.conf; \
fi

root@supervision:/nagios/download/nagios-4.3.2# a2enmod rewrite
root@supervision:/nagios/download/nagios-4.3.2# a2enmod a2enmod cgi
```

Création d'un compte « nagiosadmin » pour se connecter à la page Web Nagios.

```
root@supervision:/nagios/download/nagios-4.3.2# htpasswd -c /usr/local/nagios/etc/
htpasswd.users nagiosadmin
New password:
Re-type new password:
```

Le mot de passe choisi est « nagiosadmin », rien de bien compliqué !
On peut redémarrer le service apache2.

```
root@supervision:/nagios/download/nagios-4.3.2# /etc/init.d/apache2 stop
[ OK ] Stopping apache2 (via systemctl): apache2.service.
root@supervision:/nagios/download/nagios-4.3.2# /etc/init.d/apache2 start
[ OK ] Starting apache2 (via systemctl): apache2.service.

Ou

service apache2 restart

ou

systemctl restart apache2.service
```

II-F - Compilation et installation des plug-ins Nagios

Remettons-nous dans le répertoire de téléchargement afin de décompresser les sources, les compiler et les installer.

```
root@supervision:/nagios/download/nagios-4.3.2# cd /nagios/download/
root@supervision:/nagios/download# tar -xzf nagios-plugins-2.2.1.tar.gz
root@supervision:/nagios/download# cd nagios-plugins-2.2.1/
root@supervision:/nagios/download/nagios-plugins-2.2.1# ./configure --with-nagios-user=nagios --
with-nagios-group=nagios

root@supervision:/nagios/download/nagios-plugins-2.2.1# make
root@supervision:/nagios/download/nagios-plugins-2.2.1# make install
```

II-G - Démarrage de Nagios

À la première installation, et à chaque fois que vous ferez des modifications de Nagios ou que vous aurez besoin de redémarrer Nagios, assurez-vous que tous les fichiers de configuration sont conformes. Pour cela, lancez toujours cette commande.

```
root@supervision:/nagios/download/nagios-plugins-2.2.1# /usr/local/nagios/bin/nagios -v /usr/
local/nagios/etc/nagios.cfg

Nagios Core 4.3.2
Copyright (c) 2009-present Nagios Core Development Team and Community Contributors
```



```
Copyright (c) 1999-2009 Ethan Galstad
Last Modified: 2017-05-09
License: GPL

Website: https://www.nagios.org
Reading configuration data...
  Read main config file okay...
  Read object config files okay...

Running pre-flight check on configuration data...

Checking objects...
  Checked 8 services.
  Checked 1 hosts.
  Checked 1 host groups.
  Checked 0 service groups.
  Checked 1 contacts.
  Checked 1 contact groups.
  Checked 24 commands.
  Checked 5 time periods.
  Checked 0 host escalations.
  Checked 0 service escalations.
Checking for circular paths...
  Checked 1 hosts
  Checked 0 service dependencies
  Checked 0 host dependencies
  Checked 5 timeperiods
Checking global event handlers...
Checking obsessive compulsive processor commands...
Checking misc settings...

Total Warnings: 0
Total Errors: 0

Things look okay - No serious problems were detected during the pre-flight check
```

S'il n'y a pas d'erreur, on peut continuer.

- Gestion du démarrage du serveur

Il est important que Nagios et Apache2 puissent être lancés au démarrage du serveur. Pour configurer cela, voici ce qu'il faut faire :

Sous Suse

```
supervision:/nagios/download/nagios-plugins-2.2.1 # chkconfig --add apache2
supervision:/nagios/download/nagios-plugins-2.2.1 # chkconfig --add nagios
supervision:/nagios/download/nagios-plugins-2.2.1 # chkconfig nagios on
supervision:/nagios/download/nagios-plugins-2.2.1 # chkconfig apache2 on
```

Sous Debian, l'équivalent de « chkconfig » est « update-rc.d » et la commande est la suivante :

update-rc.d

```
# update-rc.d nagios defaults
```

Nous l'avons déjà exécuté ci-dessus, c'est en rappel en cas de soucis, car il se peut que nous obtenions des messages d'erreurs. Exemple :

Erreurs : update-rc.d

```
root@supervision:/nagios/download/nagios-plugins-2.2.1# update-rc.d nagios defaults
update-rc.d: using dependency based boot sequencing
insserv: Script nagios is broken: incomplete LSB comment.
insserv: missing `Default-Start:' entry: please add even if empty.
insserv: missing `Default-Stop:' entry: please add even if empty.
insserv: Default-Start undefined, assuming empty start runlevel(s) for script `nagios'
insserv: Default-Stop undefined, assuming empty stop runlevel(s) for script `nagios'
root@supervision:/nagios/download/nagios-plugins-2.2.1#
```

Dans le fichier « /etc/init.d/nagios », il manque deux lignes essentielles :

```
# Default-Start:    2 3 4 5
# Default-Stop:    0 1 6
```

Ces deux lignes permettent de donner le rang de démarrage et d'arrêt du script nagios. Voici notre entête du fichier avant modification :

```
shell
#!/bin/sh

# Nagios      Startup script for the Nagios monitoring daemon
#
# chkconfig:  - 85 15
# description: Nagios is a service monitoring system
# processname: nagios
# config: /etc/nagios/nagios.cfg
# pidfile: /var/nagios/nagios.pid
#
### BEGIN INIT INFO
# Provides:      nagios
# Required-Start: $local_fs $syslog $network
# Required-Stop:  $local_fs $syslog $network
# Short-Description: start and stop Nagios monitoring server
# Description:   Nagios is is a service monitoring system
### END INIT INFO

# Source function library.
. /etc/rc.d/init.d/functions
```

Faisons une sauvegarde et modifions notre fichier :

```
cp /etc/init.d/nagios /nagios/download/nagios.old
```

Voici notre entête du fichier après la modification :

```
shell
#!/bin/sh

# Nagios      Startup script for the Nagios monitoring daemon
#
# chkconfig:  - 85 15
# description: Nagios is a service monitoring system
# processname: nagios
# config: /etc/nagios/nagios.cfg
# pidfile: /var/nagios/nagios.pid
#
### BEGIN INIT INFO
# Provides:      nagios
# Required-Start: $local_fs $syslog $network
# Required-Stop:  $local_fs $syslog $network
# Default-Start: 2 3 4 5
# Default-Stop: 0 1 6
# Short-Description: start and stop Nagios monitoring server
# Description:   Nagios is is a service monitoring system
### END INIT INFO

# Source function library.
. /etc/rc.d/init.d/functions
```

Maintenant, relançons l'utilitaire « update-rc.d ».

```
root@supervision:/nagios/download/nagios-plugins-2.2.1# update-rc.d nagios defaults
update-rc.d: using dependency based boot sequencing
```

Maintenant, démarrons Nagios :

```
root@supervision:/nagios/download/nagios-plugins-2.2.1# /etc/init.d/nagios start
```

Sous Debian, il m'est arrivé d'avoir au démarrage des messages d'erreurs :

```
root@supervision:/nagios/download/nagios-plugins-2.2.1# /etc/init.d/nagios start
/etc/init.d/nagios: 22: .: Can't open /etc/rc.d/init.d/functions
```

Au moment de la mise à jour de ce tutoriel, ce bogue n'existait plus, donc probablement corrigé, néanmoins, voici ce que j'ai fait pour corriger !

J'ai commenté la ligne suivante, car sur ma Debian, ce fichier n'était pas trouvé :

```
. /etc/rc.d/init.d/functions
```

J'ai rajouté la ligne suivante « `./lib/lsb/init-functions` », ce qui donne :

```
# Source function library.
# . /etc/rc.d/init.d/functions
. /lib/lsb/init-functions
```

Cette simple modification ne suffisait malheureusement pas, on obtenait encore des messages d'erreurs :

```
root@supervision:/nagios/download/nagios-plugins-2.2.1# /etc/init.d/nagios start
/etc/init.d/nagios: 119: /etc/init.d/nagios: status: not found
/etc/init.d/nagios: 76: test: false: unexpected operator
$Starting nagios: Invalid --user argument: 'd' (unknown user d)
usage: daemon [options] [--] [cmd arg...]
options:
...
```

Voici quelques modifications apportées à l'époque au script de démarrage Nagios.

Il fallait changer la ligne :

```
/sbin/service nagios configtest > "$TMPFILE"
```

Sous ma Debian, le fichier service ne se trouve pas dans le répertoire `/sbin`, mais dans le répertoire « `/usr/sbin/` » :

```
/usr/sbin/service nagios configtest > "$TMPFILE"
```

Les lignes « `status $prog` » sont remplacées par « `status_of_proc $prog` »

Dans le `stop()`, remplacer :

```
Erreur : /etc/init.d/nagios stop
killproc -p ${pidfile} -d 10 $exec
```

Par

```
killproc -p ${pidfile} $exec
```

Remplacer :

```
Erreur : /etc/init.d/nagios configtest
snice runuser -s /bin/bash - $user -c "$corelimit >/dev/null 2>&1 ; $exec -vp $config"
```

Par :

```
$nice su -s /bin/bash - nagios -c "$corelimit >/dev/null 2>&1 ; $exec -vp $config"
```

Au niveau du start, voici deux lignes à changer :

```
if test "$checkconfig" == "false"; then  
...  
daemon --user=$user $exec -ud $config
```

par :

```
if test "$checkconfig" = "false"; then  
...  
daemon --user=$user -- $exec -d $config
```

Remplacer

```
lockfile=/var/lock/subsys/$prog
```

Par :

```
lockfile=/var/lock/$prog
```

Pour résumer, votre fichier /etc/init.d/nagios sous Debian devenait :

```
/etc/init.d/nagios corrigé  
#!/bin/sh  
  
# Nagios Startup script for the Nagios monitoring daemon  
#  
# chkconfig: - 85 15  
# description: Nagios is a service monitoring system  
# processname: nagios  
# config: /etc/nagios/nagios.cfg  
# pidfile: /var/nagios/nagios.pid  
#  
### BEGIN INIT INFO  
# Provides: nagios  
# Required-Start: $local_fs $syslog $network  
# Required-Stop: $local_fs $syslog $network  
# Default-Start: 2 3 4 5  
# Default-Stop: 0 1 6  
# Short-Description: start and stop Nagios monitoring server  
# Description: Nagios is is a service monitoring system  
### END INIT INFO  
  
# Source function library.  
# . /etc/rc.d/init.d/functions  
# . /lib/lsb/init-functions  
  
prefix="/usr/local/nagios"  
exec_prefix="${prefix}"  
exec="${exec_prefix}/bin/nagios"  
prog="nagios"  
config="${prefix}/etc/nagios.cfg"  
pidfile="${prefix}/var/nagios.lock"  
user="nagios"  
group="nagios"  
checkconfig="false"  
ramdiskdir="/var/nagios/ramcache"  
  
test -e /etc/sysconfig/$prog && . /etc/sysconfig/$prog
```

/etc/init.d/nagios corrigé

```
lockfile=/var/lock/$prog
USE_RAMDISK=${USE_RAMDISK:-0}

if test "$USE_RAMDISK" -ne 0 && test "$RAMDISK_SIZE"X != "X"; then
    ramdisk=`mount |grep "$ramdiskdir type tmpfs"`
    if [ "$ramdisk"X == "X" ]; then
        mkdir -p -m 0755 $ramdiskdir
        mount -t tmpfs -o size=${RAMDISK_SIZE}m tmpfs $ramdiskdir
        mkdir -p -m 0755 $ramdiskdir/checkresults
        chown -R $user:$group $ramdiskdir
    fi
fi

check_config() {
    TMPFILE="/tmp/.configtest.$$"
    /usr/sbin/service nagios configtest > "$TMPFILE"
    WARN=`grep ^"Total Warnings:" "$TMPFILE" |awk -F: '{print \$2}' |sed s/' '//g`
    ERR=`grep ^"Total Errors:" "$TMPFILE" |awk -F: '{print \$2}' |sed s/' '//g`

    if test "$WARN" = "0" && test "${ERR}" = "0"; then
        echo "OK - Configuration check verified" > /var/run/nagios.configtest
        chmod 0644 /var/run/nagios.configtest
        /bin/rm "$TMPFILE"
        return 0
    else
        # We'll write out the errors to a file we can have a
        # script watching for
        echo "WARNING: Errors in config files - see log for details: $TMPFILE" > /var/run/
nagios.configtest
        egrep -i "(^warning|^error)" "$TMPFILE" >> /var/run/nagios.configtest
        chmod 0644 /var/run/nagios.configtest
        cat "$TMPFILE"
        exit 8
    fi
}

start() {
    test -x $exec || exit 5
    test -f $config || exit 6
    if test "$checkconfig" = "false"; then
        check_config
    fi
    echo -n "$Starting $prog: "
    # We need to _make sure_ the precache is there and verified
    # Raise priority to make it run better
    daemon --user=$user -- $exec -d $config
    retval=$?
    echo
    test $retval -eq 0 && touch $lockfile
    return $retval
}

stop() {
    echo -n "$Stopping $prog: "
    killproc -p ${pidfile} $exec
    retval=$?
    echo
    test $retval -eq 0 && rm -f $lockfile
    return $retval
}

restart() {
    check_config
    checkconfig="true"
    stop
    start
}

reload() {
```

/etc/init.d/nagios corrigé

```

echo -n $"Reloading $prog: "
killproc -p ${pidfile} $exec -HUP
RETVAL=$?
echo
}

force_reload() {
    restart
}

case "$1" in
    start)
        status_of_proc $prog && exit 0
    $1
    ;;
    stop)
        status_of_proc $prog || exit 0
    $1
    ;;
    restart)
        $1
    ;;
    reload)
        status_of_proc $prog || exit 7
    $1
    ;;
    force-reload)
        force_reload
    ;;
    status)
        status_of_proc $prog
    ;;
    condrestart|try-restart)
        status_of_proc $prog || exit 0
        restart
    ;;
    configtest)
        $nice su -s /bin/bash - nagios -c "$corelimit >/dev/null 2>&1 ; $exec -vp $config"
        RETVAL=$?
    ;;
    *)
        echo $"Usage: $0 {start|stop|status|restart|condrestart|try-restart|reload|force-reload|configtest}"
        exit 2
esac
exit $?
    
```



Avant d'appliquer cette correction, il faut bien s'assurer que vous obteniez bien les messages d'erreurs successifs. Il se peut qu'au moment de la lecture de ce tutoriel, le bogue ait été corrigé par Nagios.

Tous ces tests et modifications ont peut-être créé des processus Nagios zombies. Un redémarrage de votre serveur fera du bien et permettra de tester que Nagios se lance et s'arrête correctement.

```

root@supervision:~# /etc/init.d/nagios stop
[ OK ] is running.
$Stopping nagios:
root@supervision:~# ps -aux | grep nagios
warning: bad ps syntax, perhaps a bogus '-'?
See http://gitorious.org/procps/procps/blobs/master/Documentation/FAQ
root      3541  0.0  0.0  3564   788 pts/0    S+   13:17   0:00 grep nagios
root@supervision:~# /etc/init.d/nagios stop
[FAIL] is not running ... failed!
root@supervision:~# /etc/init.d/nagios start
[FAIL] is not running ... failed!
$Starting nagios:
    
```

```

root@supervision:# ps -aux | grep nagios
warning: bad ps syntax, perhaps a bogus '-l'?
See http://gitorious.org/procps/procps/blobs/master/Documentation/FAQ
nagios  3630  0.0  0.1  7304  1136 ?        Ss   13:18   0:00 /usr/local/nagios/bin/nagios -
d /usr/local/nagios/etc/nagios.cfg
nagios  3631  0.0  0.0  2672  588 ?        S    13:18   0:00 /usr/local/nagios/bin/nagios --
worker /usr/local/nagios/var/rw/nagios.qh
nagios  3632  0.0  0.0  2672  588 ?        S    13:18   0:00 /usr/local/nagios/bin/nagios --
worker /usr/local/nagios/var/rw/nagios.qh
nagios  3633  0.0  0.0  2672  588 ?        S    13:18   0:00 /usr/local/nagios/bin/nagios --
worker /usr/local/nagios/var/rw/nagios.qh
nagios  3634  0.0  0.0  2672  580 ?        S    13:18   0:00 /usr/local/nagios/bin/nagios --
worker /usr/local/nagios/var/rw/nagios.qh
nagios  3635  0.0  0.0  6928  400 ?        S    13:18   0:00 /usr/local/nagios/bin/nagios -
d /usr/local/nagios/etc/nagios.cfg
root    3637  0.0  0.0  3564  792 pts/0    S+   13:18   0:00 grep nagios
root@supervision:# /etc/init.d/nagios status
[ OK ] is running.
root@supervision:#
    
```

Le script `/etc/init.d/nagios start|stop|restart` n'est pas toujours d'une fiabilité sans faille. Afin d'éviter d'avoir plusieurs démons Nagios qui tourneraient et qui mettraient à jour la même interface CGI. Voici comment éviter ce désagrément :

```

root@supervision:/nagios/download/nagios-plugins-2.2.1# /usr/local/nagios/bin/nagios -v /usr/
local/nagios/etc/nagios.cfg

root@supervision:/nagios/download/nagios-plugins-2.2.1# /etc/init.d/nagios stop; pkill nagios; /
etc/init.d/nagios start
    
```

Changer la permission de ce répertoire `/usr/local/nagios/var/rw` afin de pouvoir effectuer certaines actions depuis l'interface Nagios.

```

supervision:/nagios/download/nagios-plugins-2.2.1 # chown nagios.nagcmd /usr/local/nagios/var/rw
    
```

Avant de se connecter à la page Web de Nagios, il est important de vérifier que le pare-feu du serveur ne bloque pas l'affichage des pages Web. C'est le cas sous Suse. Pour cela, allons dans

- YaST ;
- Catégorie « Sécurité et utilisateurs » ;
- Pare-feu ;
- Rajoutons « Serveur http » à la liste des services autorisés de la zone externe.

Sous Debian, vous devez avoir le port 80 non bloqué par le firewall (c'est généralement le cas par défaut). Juste pour information si besoin :

Répondre oui

```

iptables -I INPUT -p tcp --destination-port 80 -j ACCEPT
apt-get install -y iptables-persistent
    
```

Maintenant nous pouvons nous connecter à Nagios via le lien suivant : <http://supervision/nagios>.



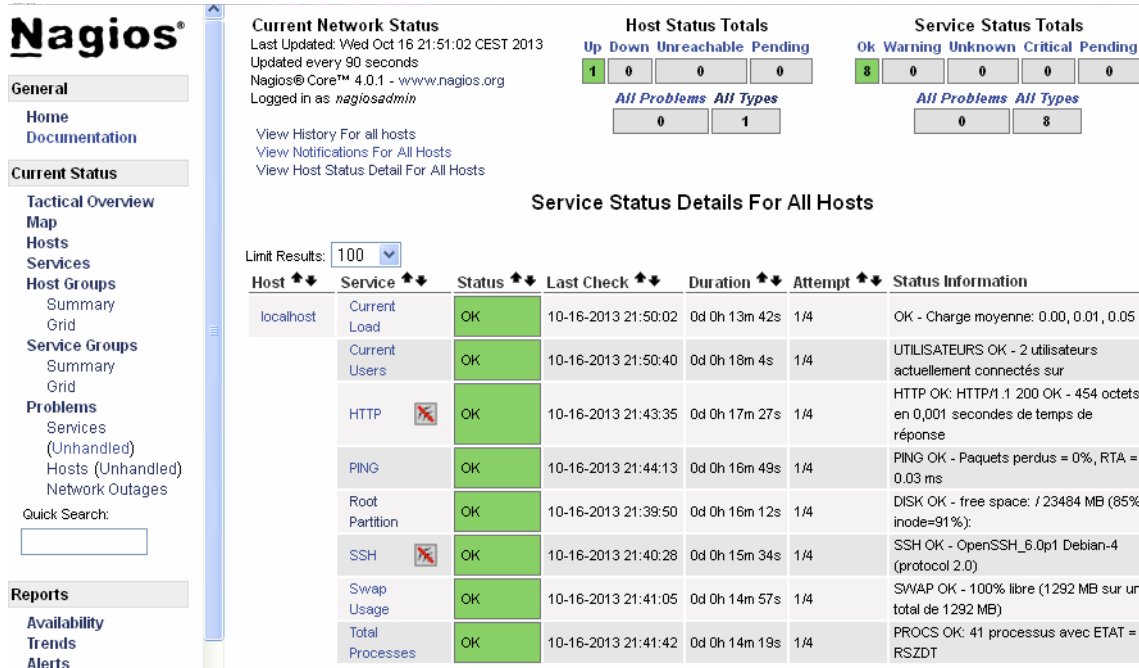
En cas de page blanche, non trouvée, voire [ici](#).

III - Configuration de Nagios

Nous n'allons pas voir comment configurer entièrement Nagios, car c'est infini et pour en savoir plus, vous devrez vous pencher sur la documentation officielle. Néanmoins, nous allons essayer de paramétrer Nagios pour surveiller l'espace disque du serveur Nagios et d'autres serveurs Windows et Linux ainsi que leur accessibilité.

III-A - Surveillance locale

À ce stade de l'article, Nagios ne surveille que le serveur sur lequel il est installé. Si vous cliquez sur le lien Services ou Hosts de la page Web de votre Nagios (<http://supervision/nagios/>), vous ne verrez que localhost. Certains services sont par défaut surveillés, notamment le swap, l'espace disque de la partition root /, le ping, etc.



Nagios
 General: Home, Documentation
 Current Status: Tactical Overview, Map, Hosts, Services, Host Groups, Service Groups, Problems, Quick Search
 Reports: Availability, Trends, Alerts

Current Network Status
 Last Updated: Wed Oct 16 21:51:02 CEST 2013
 Updated every 90 seconds
 Nagios® Core™ 4.0.1 - www.nagios.org
 Logged in as *nagiosadmin*

Host Status Totals
 Up: 1, Down: 0, Unreachable: 0, Pending: 0
 All Problems: 0, All Types: 1

Service Status Totals
 Ok: 8, Warning: 0, Unknown: 0, Critical: 0, Pending: 0
 All Problems: 0, All Types: 8

Service Status Details For All Hosts
 Limit Results: 100

Host	Service	Status	Last Check	Duration	Attempt	Status Information
localhost	Current Load	OK	10-16-2013 21:50:02	0d 0h 13m 42s	1/4	OK - Charge moyenne: 0.00, 0.01, 0.05
	Current Users	OK	10-16-2013 21:50:40	0d 0h 18m 4s	1/4	UTILISATEURS OK - 2 utilisateurs actuellement connectés sur
	HTTP	OK	10-16-2013 21:43:35	0d 0h 17m 27s	1/4	HTTP OK: HTTP/1.1 200 OK - 454 octets en 0,001 secondes de temps de réponse
	PING	OK	10-16-2013 21:44:13	0d 0h 16m 49s	1/4	PING OK - Paquets perdus = 0%, RTA = 0.03 ms
	Root Partition	OK	10-16-2013 21:39:50	0d 0h 16m 12s	1/4	DISK OK - free space: / 23484 MB (85% inode=91%)
	SSH	OK	10-16-2013 21:40:28	0d 0h 15m 34s	1/4	SSH OK - OpenSSH_6.0p1 Debian-4 (protocol 2.0)
	Swap Usage	OK	10-16-2013 21:41:05	0d 0h 14m 57s	1/4	SWAP OK - 100% libre (1292 MB sur un total de 1292 MB)
	Total Processes	OK	10-16-2013 21:41:42	0d 0h 14m 19s	1/4	PROCS OK: 41 processus avec ETAT = RSZDT

Pour modifier les services surveillés par Nagios en local, il faut aller dans le fichier localhost.cfg qui se trouve dans le répertoire /usr/local/nagios/etc/objects. Vous y trouverez différentes sections. La première est la section host. Pour la vérification de l'espace disque, il faut voir le service utilisant le programme « check_local_disk ». Vous trouverez cette ligne :

```
check_command          check_local_disk!20%!10%!/
```

En fait, Nagios lance un programme s'appelant check_local_disk en lui passant en argument 20% et 10%. Ce programme va vérifier l'espace disque et envoyer à Nagios 0 si tout est OK, 1 (pour warning) si l'espace disque restant est inférieur à 20 % ou 2 (pour critique) si l'espace disque restant est inférieur à 10 %. Avec ce code retour, Nagios va afficher un message adapté au niveau de l'interface Web. Vous pouvez les modifier pour tester.

- Comment Nagios fait-il pour connaître l'emplacement des programmes ?

En fait, il existe un fichier « commands.cfg » dans le répertoire /usr/local/nagios/etc/objects où sont listées toutes les commandes que Nagios utilisera en local.

```
Exemple
# 'check_local_disk' command definition
define command{
  command_name    check_local_disk
  command_line    $USER1$/check_disk -w $ARG1$ -c $ARG2$ -p $ARG3$ -u $ARG4$
}
```

On remarquera que Nagios lance en réalité le programme check_disk au lieu de check_local_disk qui est en fait un alias avec les paramètres -w \$ARG1\$ -c \$ARG2\$ -p \$ARG3\$. \$ARG1 correspond au 20%, \$ARG2 au 10% et \$ARG3 à /. Pour en savoir plus sur les arguments, consultez la documentation officielle [nagiosplugins](#) et cherchez check_disk.

Mettons-nous à la place de Nagios en ligne de commande et lançons `check_disk`. À noter que tous les programmes Nagios se trouvent dans le répertoire `/usr/local/nagios/libexec`.

```
supervision:~ # /usr/local/nagios/libexec/check_disk -w 20% -c 10% -p /
DISK OK - free space: / 137429 MB (97% inode=98%); | /=3734MB;118974;133846;0;148718
supervision:~ # echo $?
0
```

Vous remarquez que le code retourné est bien 0. Affinons notre requête pour afficher les tailles en Gigabytes au lieu de MB (par défaut). L'option `-u` nous le permet.

```
supervision:~ # /usr/local/nagios/libexec/check_disk -w 20% -c 10% -p / -u GB
DISK OK - free space: / 134 GB (97% inode=98%); | /=3GB;116;130;0;145
```

Si vous souhaitez que Nagios fasse le même affichage, il faut modifier le fichier « `commands.cfg` » pour soit lui mettre en dur `-u GB`, soit lui donner un quatrième argument nous donnant le choix de le paramétrer. Exemple :

```
supervision:~ # cat /usr/local/nagios/etc/objects/commands.cfg | grep check_disk
command_line $USER1$/check_disk -w $ARG1$ -c $ARG2$ -p $ARG3$ -u $ARG4$
supervision:~ # cat /usr/local/nagios/etc/objects/localhost.cfg | grep check_local_disk
check_command check_local_disk!20%!10%!/!GB
```

Voilà, c'est magique !!! Pour les autres services, le principe est le même. Avec un peu, voire beaucoup d'huile de coude et vous pourrez y arriver 😊!!

III-B - Surveillance à distance

Pour surveiller des machines distantes, il faut qu'il y ait sur ces machines un agent qui pourra renseigner les plug-ins Nagios des informations dont ils ont besoin.



Par surveillance, je sous-entends supervision d'attributs et services privés (CPU, espace disque...). Pour une supervision de services publics (HTTP, FTP...), il n'y a pas forcément besoin d'agents.

La mise en place de cet agent diffère en fonction du système d'exploitation de la machine à surveiller ou du matériel. Nous prendrons pour exemple une machine Windows, Linux et un switch. Une fois l'agent installé, il faut créer de nouvelles définitions d'hôtes et de services pour surveiller la machine puis redémarrer le démon Nagios.

III-B-1 - Machines à surveiller sous Windows

Pour notre tutoriel, nous allons installer l'addon NSClient++ sur la machine Windows et utiliser le greffon `check_nt` pour communiquer avec NSClient++.



« NSClient++, notamment la version 0.3.9 est la dernière version mise à disposition du public à la date du 02/04/2012. À partir de mai 2012, il faudra installer NSCP comme agent sur vos machines Windows si vous souhaitez être à jour. Bien évidemment, si vous souhaitez toujours utiliser NSClient++, il sera toujours fonctionnel, voire disponible en téléchargement sur le site officiel.

Ce greffon `check_nt` est déjà installé vu que Nagios l'est. Vous pouvez le trouver dans le fichier « `/usr/local/nagios/etc/objects/commands.cfg` ».

Dans /usr/local/nagios/etc/objects/commands.cfg

```
# 'check_nt' command definition
define command{
    command_name     check_nt
    command_line     $USER1$/check_nt -H $HOSTADDRESS$ -s <mot-de-passe> -p 12489 -v $ARG1$ $ARG2$
}
```



Il se peut que vous n'avez pas la chaîne -s <mot-de-passe> dans votre fichier de configuration. Veuillez la rajouter et changer « <mot-de-passe> » par un mot de passe de votre choix qui sera utilisé pour surveiller les machines Windows.

Voyons comment installer l'agent Windows.

III-B-1-a - Agent NSClient++ jusqu'en avril 2012

Il est possible d'utiliser d'autres agents Windows (comme NC_Net), mais il faudra dans ce cas modifier les commandes et les définitions de services... Mais nous, nous utiliserons NSClient++. Téléchargez-le à cette [adresse \(old legacy versions\)](#) et récupérez NSClient++-0.3.9-Win32.msi ou une version plus récente. À ce jour (16/10/2013), c'est la version 0.3.9. Je vous rappelle que notre machine Windows s'appelle « winprod ».

- Installation de l'agent

Une fois votre fichier MSI téléchargé, installez-le sur la machine winprod. Il vous demandera l'adresse IP de votre serveur Nagios, vous pouvez mettre le nom DNS si votre réseau en dispose d'un. Précisez un mot de passe, mettez-le dans le champ adéquat. Ensuite activez les champs plug-ins, check_nt et check_nrpe.

NSClient++ (Win32)

NSClient++ Configuration

Old configuration was not found!

Allowed hosts: (this is the IP of the nagios (or other) server)

NSClient password (only used via check_nt):

Modules to load:

- Enable common check plugins
- Enable nsclient server (check_nt)
- Enable NRPE server (check_nrpe)
- Enable NSCA client (dont enable unless you really use NSCA)
- Enable WMI checks

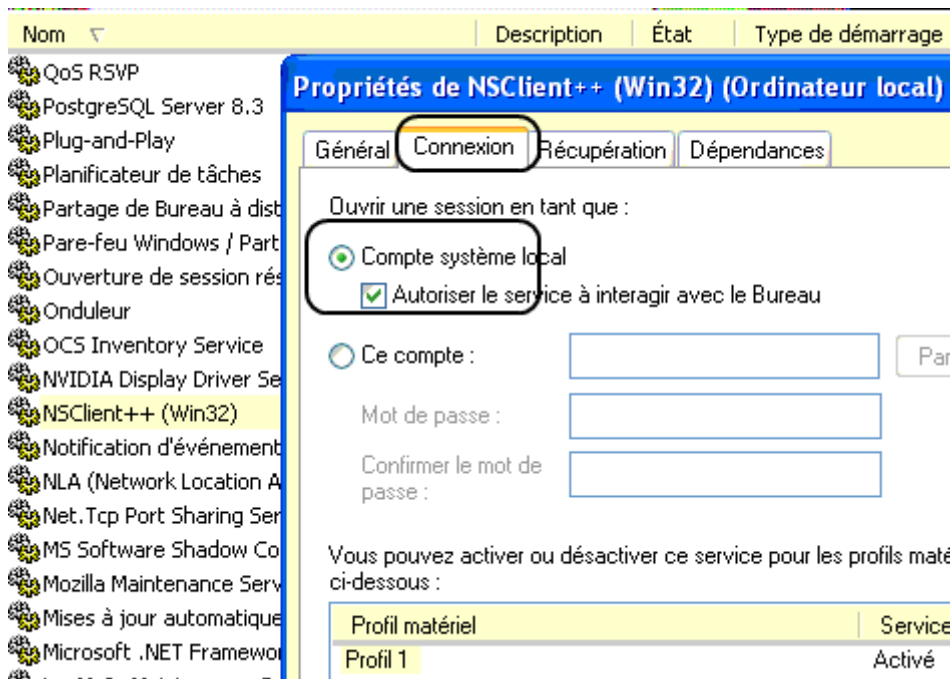
Back Next



Si vous ne mettez pas de mot de passe, il se peut que plus tard, vous ayez ce genre de message d'erreur : « NSClient - ERROR: Invalid password. ».

Le mot de passe peut être par la suite modifié après l'installation de NSClient en modifiant le fichier « NSC.ini » qui se trouve à cet emplacement : C:\Program Files\NSClient++\ (ou C:\Program\NSClient++\ pour une version 64 bits).

Ouvrez maintenant le gestionnaire de service afin de s'assurer que le service NSClient est autorisé à interagir avec le bureau :



Maintenant, il faut ouvrir le fichier « NSC.ini » qui se trouve à cet emplacement : C:\Program Files\NSClient++\ (ou C:\Program\NSClient++\ pour une version 64 bits).

Dans la section [modules], il faut décommenter tous les modules listés exceptés CheckWMI.dll et RemoteConfiguration.dll.

Dans la section [Settings], vous devez avoir ces deux lignes :

```
password=VOTRE MOT DE PASSE
allowed_hosts=supervision
```

III-B-1-b - Agent NSCP à partir de mai 2012

NSCP n'est autre chose que NSClient++ amélioré. C'est l'agent à utiliser depuis mai 2012. Sa configuration diffère un peu de celle de NSClient++ et on ne trouve pas beaucoup de documentations à ce jour en dehors de l'officielle pas toujours très compréhensible. Néanmoins, dans ce tutoriel, je vous présenterai une de mes configurations m'ayant permis d'utiliser Nagios.

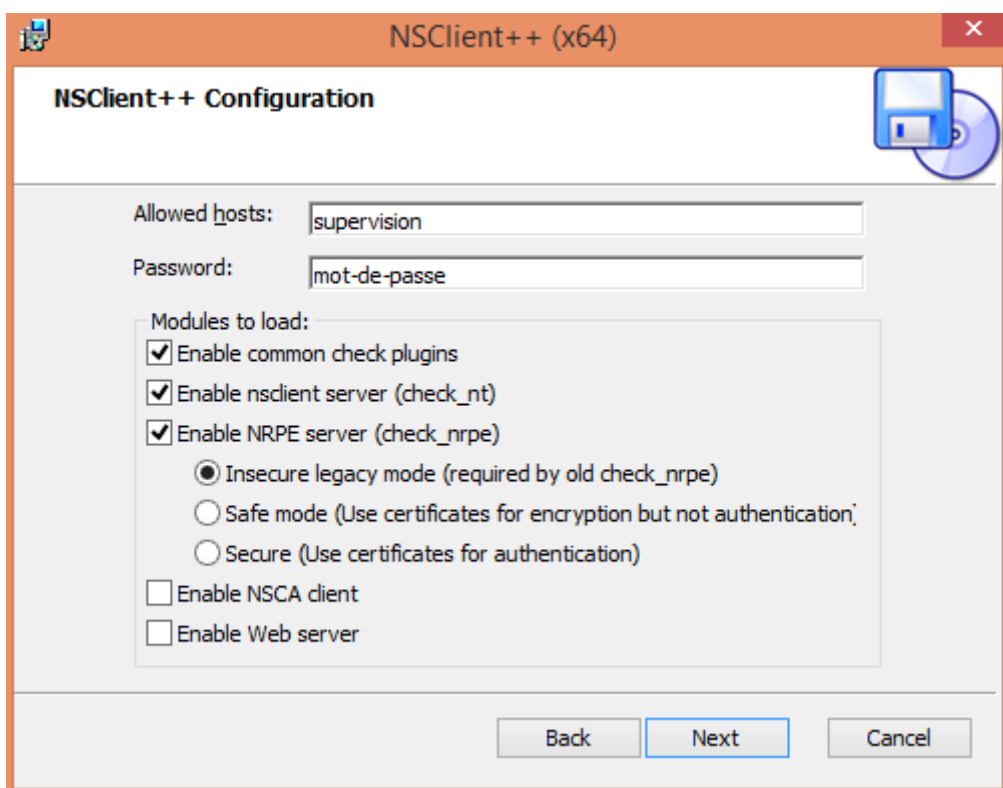
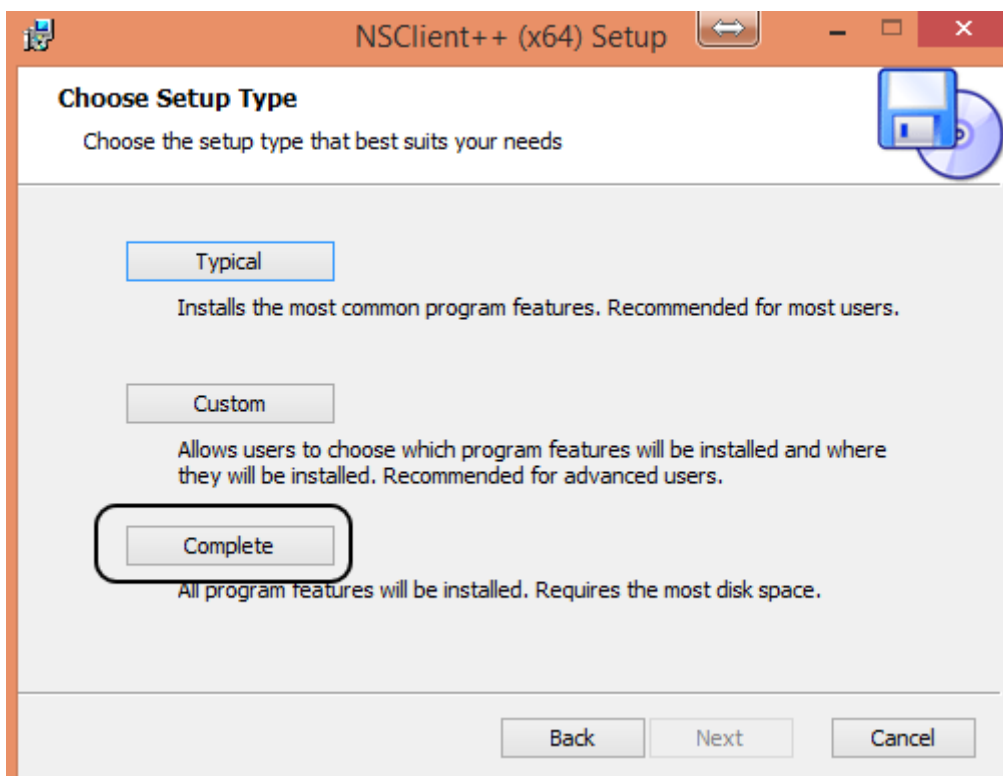
- Installation de l'agent

Téléchargez-le à cette [adresse](#) et récupérez NSCP-0.4.4.19 (x64) (nous sommes sur une machine 64 bits) ou une version plus récente. À ce jour (12/04/2016), c'est la version 0.4.4.19.

Je vous rappelle que notre machine Windows s'appelle « winprod ».


Une fois votre fichier MSI téléchargé, installez-le sur la machine winprod. Il vous demandera l'adresse IP de votre serveur Nagios, vous pouvez mettre le nom DNS si votre réseau en dispose d'un.

Choisissez une installation complète et précisez un mot de passe, mettez-le dans le champ adéquat. Ensuite activez les champs plug-ins, check_nt et check_nrpe.



Si vous ne mettez pas de mot de passe, il se peut que plus tard, vous ayez ce genre de message d'erreur : « NSClient - ERROR: Invalid password. ».

Le mot de passe peut être par la suite modifié après l'installation de NSClient en modifiant le fichier « nsclient.ini » qui se trouve à cet emplacement : C:\Program Files\NSClient++.

 *Je vous recommande d'ouvrir le fichier « nsclient.ini » en tant qu'administrateur. Par exemple, en ouvrant notepad++ en tant qu'administrateur, puis vous ouvrez votre fichier. Sinon, il se peut que vous ne puissiez pas sauvegarder votre fichier faute d'autorisation.*

Dans votre fichier « nsclient.ini », s'il n'existe pas de section [/settings/external scripts], veuillez en rajouter une en fin de fichier de la sorte :

```
Autoriser l'ajout d'arguments
[/settings/external scripts]
allow arguments = true
```

Dans la section [/settings/NRPE/server], veuillez en rajouter la ligne :

```
Autoriser l'ajout d'arguments
allow arguments = true
```

Supposons que nous avons développé un plug-in Perl que nous souhaitons exécuter sous Windows, pour qu'il fonctionne, voici ce dont vous avez besoin :

- **Installer Perl** ;
- Chercher la section [/settings/external scripts/scripts] dans le .ini ;
- Rajouter une ligne comme ci-dessous :

```
Ligne d'appel d'un programme Perl
[/settings/external scripts/scripts]


; SCRIPT - For more configuration options add a dedicated section (if you add a new section you can customize the
foo = scripts\fofo.bat

check_ntp=C:\Perl64\bin\perl.exe scripts\perl\check_ntp.pl -H $ARG1$ -w $ARG2$ -c $ARG3$
```

Dans l'exemple ci-dessus, on a créé un programme Perl nommé « check_ntp.pl » qui accepte trois arguments. (-H, -w et -c). Ce programme sera appelé depuis le serveur Nagios via l'alias « check_ntp ». Pour qu'il soit lancé via perl, on donne le chemin complet de l'exécutable perl.

N.B. Vous noterez également que l'on a créé un répertoire C:\Program Files\NSClient++\scripts\perl afin d'y mettre notre plug-in. Pour en savoir plus sur la création de plugin Perl : [Concevoir facilement un plug-in Nagios en Perl](#).

À chaque mise à jour du fichier de configuration « nsclient.ini », vous devez redémarrer le service nscp. Vous pouvez le faire en ligne de commande depuis une console DOS ouverte en tant qu'administrateur de la sorte :

 **Commande DOS**

```
cd "C:\Program Files\NSClient++"
net stop nscp && net start nscp
```

III-B-1-c - Configuration de Nagios

Remettons-nous sur notre serveur Nagios « supervision » et éditons le fichier de configuration de Nagios.

```
supervision:~ # vi /usr/local/nagios/etc/nagios.cfg

# Definitions for monitoring a Windows machine
#cfg_file=/usr/local/nagios/etc/objects/windows.cfg
```

Si nous décommentons la ligne avec windows.cfg, nous disons à Nagios de regarder le fichier windows.cfg pour y trouver les définitions des hôtes Windows. Cette méthode est intéressante, car nous pouvons définir toutes nos machines Windows dans le même fichier, mais je préfère avoir un fichier par serveur Windows pour une meilleure lisibilité et maintenance. Pour ce faire, je n'utiliserai pas cette technique de configuration.

Créons un répertoire où stocker tous les fichiers de configuration de nos machines Windows et précisons-le à Nagios. Mettons-nous sous l'utilisateur « nagios ».

```
supervision:~ # su - nagios
nagios@supervision:~$ mkdir /usr/local/nagios/etc/serveurs_windows
nagios@supervision:~$ vi /usr/local/nagios/etc/nagios.cfg
```

Trouvons les lignes commentées commençant par #cfg_dir=... Créons une ligne identique en précisant le chemin vers le répertoire que nous avons créé. Décommentez-la.

Voici le résultat :

```
...
cfg_dir=/usr/local/nagios/etc/serveurs_windows
#cfg_dir=/usr/local/nagios/etc/servers
#cfg_dir=/usr/local/nagios/etc/printers
#cfg_dir=/usr/local/nagios/etc/switches
#cfg_dir=/usr/local/nagios/etc/routers
```

Copions le fichier windows.cfg en winprod.cfg dans ce répertoire.

```
nagios@supervision:~$ cp /usr/local/nagios/etc/objects/windows.cfg /usr/local/nagios/etc/
serveurs_windows/winprod.cfg
```

Modifions ce fichier en remplaçant winserver par winprod (nom de notre serveur Windows). Puis changeons l'adresse IP par la nôtre ou par winprod étant donné que nous disposons d'un DNS.

À ce stade, tout devrait bien fonctionner, faisons une petite vérification !

```
nagios@supervision:~$ /usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg

Nagios Core 4.3.2

...

Total Warnings: 0
Total Errors: 0

Things look okay - No serious problems were detected during the pre-flight check
```

Dans le fichier windows.cfg, il y a une section où l'on définit tous les serveurs du groupe. Comme le but est d'avoir un fichier par machine, je ne souhaite pas devoir mettre à jour cette section dans chaque fichier de configuration pour chaque poste Windows. Créons un fichier « groupe_serveurs_windows.cfg » dans lequel on définira notre groupe de serveurs Windows. Ainsi, à chaque rajout de serveurs Windows, on fera la modification dans ce fichier.

Voici son contenu :

```
groupe_serveurs_windows.cfg
# Définition du groupe
```

groupe_serveurs_windows.cfg

```
define hostgroup{
    hostgroup_name windows-servers ; The name of the hostgroup
    alias           Serveurs Windows; Long name of the group
    members        winprod          ; separees par des virgules
}
```

Et dans notre fichier winprod.cfg, on supprime cette section. Maintenant, vérifions que notre configuration est correcte.

```
nagios@supervision:~$ /usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg
```

Il n'y a pas d'erreur, tout fonctionne bien, on peut redémarrer Nagios (sous root) et vérifier sur notre site Nagios <http://supervision/nagios>.

Dans le fichier winprod.cfg, nous testons plusieurs services présents par défaut. Notre serveur windows appartient au hostgroup « windows-servers », ce groupe doit être défini une seule fois pour toutes les machines Windows que nous surveillerons. Comme nous souhaitons faire un fichier par serveur, nous définirons ce hostgroup dans un fichier à part et l'enlèverons du fichier winprod.cfg.

```
supervision:~ # vi /usr/local/nagios/etc/serveurs_windows/hostgroup_windows.cfg

# Define a hostgroup for Windows machines
# All hosts that use the windows-server template will automatically be a member of this group

define hostgroup{
    hostgroup_name windows-servers ; The name of the hostgroup
    alias           Serveurs Windows ; Long name of the group
    members winprod ; Serveurs membres du groupe séparés par des virgules
}
```

Voilà, à ce stade, Nagios surveille notre machine winprod et vous pouvez rajouter d'autres serveurs Windows (machine.cfg dans le répertoire serveurs_windows).

III-B-2 - Machines à surveiller sous Linux

Nous avons besoin d'un agent sur les serveurs à surveiller et des plug-ins Nagios. Nous utiliserons les plug-ins « NRPE ».

NRPE (Nagios Remote Plugin Executor)(**N**agios **R**emote **P**lugin **E**xecutor) est un « Addons » pour Nagios qui permet d'exécuter des plug-ins sur un serveur Linux/Unix distant. Cela permet de surveiller des ressources locales (charge du processeur, utilisation de la mémoire, espace disque...) qui ne sont normalement pas disponibles depuis d'autres machines. Afin d'interroger le client NRPE, il faudra utiliser le greffon « check_nrpe » sur notre serveur Nagios « supervision ». Pour ce faire, il va falloir installer NRPE sur notre serveur Nagios, et sur toutes les machines à surveiller. De plus il faudra installer les plug-ins Nagios sur chaque serveur distant à surveiller.

III-B-2-a - Installation de NRPE et des plug-ins Nagios sur le serveur Linux distant

Avant de commencer l'installation, nous devons d'abord nous assurer de disposer de la bibliothèque « libssl-dev », sinon, il faut l'installer. Elle correspond à « openssl-devel » et « openssl ». Si vous avez un souci, lisez [cette aide](#).

Notre serveur servprod à surveiller est un serveur sous « Debian ». Pour les autres distributions, il faudra installer la bibliothèque « build-essential ». Il peut être également intéressant d'installer les bibliothèques make, gcc, Net-SNMP et xinetd.

Sous Debian

```
apt-get update
apt-get upgrade
apt-get install autoconf bc gawk dc libc6 make gcc libmccrypt-dev build-essential wget libssl-dev
snmpd snmp scli xinetd snmp libnet-snmp-perl gettext
```

Sous Red Hat

```
yum install openssl openssl-devel
yum install xinetd
```

- Téléchargement des archives

Plaçons-nous dans le répertoire `/usr/local/src` et téléchargeons les fichiers nécessaires (plug-in **nrpe** et **nagios**).



Si vous faites une mise à jour, il est important de faire une sauvegarde du fichier `/usr/local/nagios/etc/nrpe.cfg` !

À ce jour, nous installons NRPE version 3.1.1 et les plug-ins nagios version 2.1.1.

En cas de mise à jour ou non, si vous avez utilisé sur votre serveur nagios nrpe version 3.X comme dans ce tutoriel, vos machines clientes surveillées doivent également avoir les mêmes versions nrpe sous peine d'avoir des messages d'erreur de type : « erreur de segmentation » à la commande :

```
# /usr/local/nagios/libexec/check_nrpe -H servprod
Erreur de segmentation
```

De plus, si les machines supervisées ne sont pas à jour, le service « xinetd » ne fonctionnera pas bien.

```
# /etc/init.d/xinetd status
• xinetd.service - LSB: Starts or stops the xinetd daemon.
  Loaded: loaded (/etc/init.d/xinetd)
  Active: active (running) since sam. 2017-06-10 17:31:10 CEST; 22min ago
  Process: 4015 ExecStop=/etc/init.d/xinetd stop (code=exited, status=0/SUCCESS)
  Process: 4020 ExecStart=/etc/init.d/xinetd start (code=exited, status=0/SUCCESS)
  CGroup: /system.slice/xinetd.service
          └─4029 /usr/sbin/xinetd -pidfile /run/xinetd.pid -stayalive -
            inetd_compat -inetd_ipv6

juin 10 17:39:42 servprod nrpe[4140]: Error: Request packet type/version was
invalid!
juin 10 17:40:40 servprod nrpe[4145]: Error: Request packet type/version was
invalid!
juin 10 17:43:52 servprod nrpe[4151]: Error: Request packet type/version was
invalid!
juin 10 17:44:50 servprod nrpe[4157]: Error: Request packet type/version was
invalid!
juin 10 17:45:48 servprod nrpe[4162]: Error: Request packet type/version was
invalid!
juin 10 17:47:45 servprod nrpe[4167]: Error: Request packet type/version was
invalid!
juin 10 17:48:44 servprod nrpe[4173]: Error: Request packet type/version was
invalid!
juin 10 17:49:42 servprod nrpe[4178]: Error: Request packet type/version was
invalid!
juin 10 17:50:40 servprod nrpe[4183]: Error: Request packet type/version was
invalid!
juin 10 17:50:47 servprod nrpe[4188]: Error: Request packet type/version was
invalid!
```

Au fur et à mesure des requêtes, les lignes d'erreurs augmentent jusqu'à l'arrêt du service « nrpe ».

Il est donc préférable d'être à jour au niveau des serveurs clients, idem pour les plug-ins nagios, ça ne fera pas de mal à vos serveurs.

```
servprod:~ # cd /usr/local/src/  
servprod:/usr/local/src # wget http://downloads.sourceforge.net/project/nagios/nrpe-3.x/  
nrpe-3.1.1/nrpe-3.1.1.tar.gz  
servprod:/usr/local/src # wget http://nagios-plugins.org/download/nagios-plugins-2.1.1.tar.gz
```

- Création d'un compte et groupe nagios

Suse

```
servprod:/usr/local/src # groupadd nagios && useradd nagios -G nagios -p nagios
```

Sous Debian ou RedHat

```
servprod:/usr/local/src # groupadd nagios && useradd nagios -g nagios -p nagios
```

- Décompressons les archives

```
servprod:/usr/local/src # tar -xzf nagios-plugins-2.1.1.tar.gz && tar -xzf nrpe-3.1.1.tar.gz
```

- Compilation et installation de nagios-plugins

```
servprod:/usr/local/src # cd nagios-plugins-2.1.1/  
servprod:/usr/local/src/nagios-plugins-2.1.1 # ./configure  
servprod:/usr/local/src/nagios-plugins-2.1.1 # make && make install
```

- Compilation et installation de NRPE

```
servprod:/usr/local/src # cd /usr/local/src/nrpe-3.1.1  
servprod:/usr/local/src # ./configure --enable-command-args
```

« --enable-command-args » : permet d'activer le support des arguments des commandes.

Si vous obtenez ce message :

```
checking for SSL libraries... configure: error: Cannot find ssl libraries
```

Il faut effectuer une petite modification, car le programme ne trouve pas la bibliothèque SSL.

Sous ma Debian, j'ai recherché le fichier libssl.so.

```
# updatedb  
# locate libssl.so  
/usr/lib/i386-linux-gnu/libssl.so  
/usr/lib/i386-linux-gnu/libssl.so.1.0.0  
/usr/lib/i386-linux-gnu/i586/libssl.so.1.0.0  
/usr/lib/i386-linux-gnu/i686/cmov/libssl.so.1.0.0  
/usr/lib/vmware-tools/lib32/libssl.so.0.9.8  
/usr/lib/vmware-tools/lib32/libssl.so.0.9.8/libssl.so.0.9.8  
/usr/lib/vmware-tools/lib64/libssl.so.0.9.8  
/usr/lib/vmware-tools/lib64/libssl.so.0.9.8/libssl.so.0.9.8
```

Ensuite, j'ai créé un lien symbolique :

```
ln -s /usr/lib/i386-linux-gnu/libssl.so /usr/lib/libssl.so
```

Maintenant, lorsque je relance la commande configure, je n'ai plus d'erreur.

```

checking for socklen_t... yes
checking for type of socket size... size_t
checking for SSL headers... SSL headers found in /usr
checking for SSL libraries... SSL libraries found in /usr/lib

*** Generating DH Parameters for SSL/TLS ***
Generating DH parameters, 512 bit long safe prime, generator 2
This is going to take a long time
.....+.....+.....+.....+.....+.....+.....
+.....+.....+.....+.....+.....+.....+.....
+.....+.....+.....+.....+.....+.....+.....
+.....+.....+.....+.....+.....+.....+.....
+.....+.....+.....+.....+.....+.....+.....
+.....+.....+.....+.....+.....+.....+.....
+.....+.....+.....+.....+.....+.....+.....
+.....+.....+.....+.....+.....+.....+.....
+.....+.....+.....+.....+.....+.....+.....
+.....+.....+.....+.....+.....+.....+.....
+.....+.....+.....+.....+.....+.....+.....
+.....+.....+.....+.....+.....+.....+.....
+.....+.....+.....+.....+.....+.....+.....
checking for Kerberos include files... could not find include files
checking for perl... /usr/bin/perl
configure: creating ./config.status
config.status: creating Makefile
config.status: creating subst
config.status: creating src/Makefile
config.status: creating package/solaris/Makefile
config.status: creating init-script
config.status: creating init-script.debian
config.status: creating init-script.suse
config.status: creating nrpe.spec
config.status: creating sample-config/nrpe.cfg
config.status: creating sample-config/nrpe.xinetd
config.status: creating include/config.h

*** Configuration summary for nrpe 2.15 09-06-2013 ***:

General Options:
-----
NRPE port:      5666
NRPE user:     nagios
NRPE group:    nagios
Nagios user:   nagios
Nagios group:  nagios

Review the options above for accuracy. If they look okay,
type 'make all' to compile the NRPE daemon and client.
```

Sous Suse, il m'est arrivé d'avoir également le même souci, voire une erreur sur la recherche de SSL headers : « checking for SSL headers... configure: error: Cannot find ssl headers. », pourtant, j'avais SSL 0.9.8 et j'avais créé le lien symbolique dessus.

```

SSL : recherche SSL
# find / -name libssl.so*
/usr/lib/libssl.so.0.9.7
/usr/lib/libssl.so.0.9.8
# ln -s /usr/lib/libssl.so.0.9.8 /usr/lib/libssl.so
```

N'ayant plus le CD d'installation, j'ai téléchargé les deux RPM openssl-0.9.8a-18.36.i586.rpm et openssl-devel-0.9.8a-18.36.i586.rpm sur la toile en fouillant un peu. Tout dépend de votre version de Suse.



Si vous avez un souci d'installation de « libopenssl-devel », [cette aide](#) peut vous dépanner.

Ensuite, je les ai installés et les compilations ont fonctionné.

Suse : installation RPM

```
# rpm -ivh /tmp/openssl-devel-0.9.8a-18.36.i586.rpm
rpm -ivh /tmp/openssl-0.9.8a-18.36.i586.rpm
```

Si pour une raison quelconque vous ne pouvez pas installer SSL ou n'arrivez pas à la trouver, vous pouvez désactiver son utilisation à la compilation :

Compilation NRPE sans SSL

```
servprod:/usr/local/src # ./configure --disable-ssl --enable-command-args
```

Je vous recommande tout de même de faire le nécessaire pour l'installer.

Compilation

```
# make all
cd ./src/; make ; cd ..
make[1]: entrant dans le répertoire « /usr/local/src/nrpe-3.1.1/src »
gcc -g -O2 -I/usr/include/openssl -I/usr/include -DHAVE_CONFIG_H -I ../include -I ../include -o
nrpe ./nrpe.c ./utils.c ./acl.c -L/usr/lib -lssl -lcrypto -lnsl
gcc -g -O2 -I/usr/include/openssl -I/usr/include -DHAVE_CONFIG_H -I ../include -I ../include -o
check_nrpe ./check_nrpe.c ./utils.c -L/usr/lib -lssl -lcrypto -lnsl
make[1]: quittant le répertoire « /usr/local/src/nrpe-3.1.1/src »

*** Compile finished ***

If the NRPE daemon and client compiled without any errors, you
can continue with the installation or upgrade process.

Read the PDF documentation (NRPE.pdf) for information on the next
steps you should take to complete the installation or upgrade.
```

Installons maintenant les plug-ins NRPE :

Installation NRPE des binaires

```
root@servprod:/usr/local/src/nrpe-3.1.1# make install-plugin
cd ./src/ && make install-plugin
make[1]: entrant dans le répertoire « /usr/local/src/nrpe-3.1.1/src »
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/libexec
/usr/bin/install -c -m 775 -o nagios -g nagios check_nrpe /usr/local/nagios/libexec
make[1]: quittant le répertoire « /usr/local/src/nrpe-3.1.1/src »
```



Si vous effectuez une mise à jour, vous pouvez éviter de lancer la commande ci-dessous. Cela a pour effet de ne pas écraser votre fichier de configuration nrpe.cfg.

Installation des fichiers de configuration

```
root@servprod:/usr/local/src/nrpe-3.1.1# make install-config
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/etc
/usr/bin/install -c -m 644 -o nagios -g nagios sample-config/nrpe.cfg /usr/local/nagios/etc
```

Installation des services et démons sous différents OS :

Installation des services et démons - debian 7.x

```
root@servprod:/usr/local/src/nrpe-3.1.1# make install-init
```

Installation des services et démons - debian 7.x

```
root@servprod:/usr/local/src/nrpe-3.1.1# update-rc.d nrpe defaults
```

Installation des services et démons - debian 8.x

```
root@servprod:/usr/local/src/nrpe-3.1.1# make install-init
root@servprod:/usr/local/src/nrpe-3.1.1# systemctl enable nrpe.service
```

Installation des services et démons - SUSE SLES 11.x

```
root@servprod:/usr/local/src/nrpe-3.1.1# sudo make install-init
root@servprod:/usr/local/src/nrpe-3.1.1# sudo /sbin/chkconfig --set nrpe on
```

Installation des services et démons - SUSE SLES 12.x | openSUSE

```
root@servprod:/usr/local/src/nrpe-3.1.1# sudo make install-init
root@servprod:/usr/local/src/nrpe-3.1.1# sudo systemctl enable nrpe.service
```

Installation des services et démons - CentOS 5.x / 6.x | RHEL 5.x / 6.x | Oracle Linux 5.x / 6.x

```
root@servprod:/usr/local/src/nrpe-3.1.1# make install-init
```

Installation des services et démons - CentOS 7.x | RHEL 7.x | Oracle Linux 7.x | Fedora 23

```
root@servprod:/usr/local/src/nrpe-3.1.1# make install-init
root@servprod:/usr/local/src/nrpe-3.1.1# systemctl enable nrpe.service
```

- Modification du fichier `/etc/xinetd.d/nrpe` pour définir le service

Fichier `/etc/xinetd.d/nrpe`

```
servprod:/usr/local/src/nagios-plugins-2.2.1 # vi /etc/xinetd.d/nrpe

service nrpe
{
    flags          = REUSE
    socket_type    = stream
    port           = 5666
    wait          = no
    user           = nagios
    group          = nagios
    server         = /usr/local/nagios/bin/nrpe
    server_args    = -c /usr/local/nagios/etc/nrpe.cfg --inetd
    log_on_failure += USERID
    disable        = no
    only_from      = 127.0.0.1 192.168.1.12
}
```

À la place de `192.168.1.12`, vous pouvez mettre le nom DNS « `HOSTNAME.domaine` ».

`only_from` : permet de lister les hôtes autorisés à contacter NRPE.

Vous pouvez rajouter avant le `-c`, l'option « `-n` » qui désactive SSL (si vous n'utilisez pas SSL).

- Modifions le fichier « `/etc/services` » et rajoutons la ligne :

```
nrpe          5666/tcp          # Nagios Remote Plugin Executor
```

Il se peut qu'elle soit déjà présente.

N.B. Si vous supervisez une machine protégée par un pare-feu, pensez à ouvrir le port 5666 en TCP qui est utilisé par NRPE. Sinon vous aurez ce type de message d'erreur : ***CHECK_NRPE: Socket timeout after 10 seconds.***

- Modifions le fichier de configuration pour activer la prise d'arguments

```
servprod:/usr/local/src/nagios-plugins-2.2.1 # vi /usr/local/nagios/etc/nrpe.cfg
dont_blame_nrpe=1
```

- Redémarrage de xinetd :

```
/etc/init.d/xinetd restart
```

Sous RedHat

```
# /etc/init.d/xinetd restart
-bash: /etc/init.d/xinetd: Aucun fichier ou dossier de ce type
# /sbin/service xinetd restart
Redirecting to /bin/systemctl restart xinetd.service
```

- On teste que NRPE est bien à l'écoute et ouvert :

```
# netstat -at | grep nrpe
tcp        0      0 *:nrpe          *:*             LISTEN
```

- Effectuons des tests

Effectuons un test localement

```
/usr/local/nagios/libexec/check_nrpe -H 127.0.0.1
```

Si nous obtenons ce message d'erreur

```
CHECK_NRPE: Received 0 bytes from daemon. Check the remote server logs for error messages.
```

Vérifions les logs /var/log/messages ou /var/log/syslog. Si vous obtenez ce message d'erreur :

```
# tail -f /var/log/messages
Jan 17 16:36:27 servprod nrpe[13105]: Unable to open config file '/usr/local/nagios/etc/nrpe.cfg'
for reading
Jan 17 16:36:27 servprod nrpe[13105]: Config file '/usr/local/nagios/etc/nrpe.cfg' contained
errors, aborting...
```

Il est impossible de lire le fichier, il faut vérifier les droits

```
ls -alh /usr/local/nagios/etc/nrpe.cfg
-rw----- 1 root root 7.1K Jan 17 16:32 /usr/local/nagios/nrpe.cfg
```

En effet, Nagios ne peut pas lire le fichier, changeons les droits du fichier.

```
chmod 664 /usr/local/nagios/etc/nrpe.cfg
chown nagios.nagios /usr/local/nagios/etc/nrpe.cfg
```

Effectuons un nouveau test

```
/usr/local/nagios/libexec/check_nrpe -H 127.0.0.1
NRPE v2.15
```

Super ! Effectuons un autre test afin de compter le nombre de processus.

```
servprod:/usr/local/src/nagios-plugins-2.2.1 # /usr/local/nagios/libexec/check_nrpe -
H 127.0.0.1 -c check_total_procs
PROCS OK: 93 processes
```

Tout fonctionne parfaitement, paramétrons NRPE afin de pouvoir lire l'espace disque.

```
vi /usr/local/nagios/etc/nrpe.cfg
```

Il faut décommenter la ligne ci-dessous

```
command[check_disk]=/usr/local/nagios/libexec/check_disk -w $ARG1$ -c $ARG2$ -p $ARG3$
```

Nouveau test via check_nrpe et check_disk.

check_disk via nrpe

```
/usr/local/nagios/libexec/check_nrpe -H 127.0.0.1 -c check_disk -a 50% 100% /
DISK CRITICAL - free space: / 227593 MB (96% inode=-); | /=9373MB;118483;0;0;236967
```

L'option -a permet de passer des paramètres

check_disk directement

```
/usr/local/nagios/libexec/check_disk -w 10% -c 5% -u GB
DISK OK - free space: / 222 GB (96% inode=-); /dev 0 GB (99% inode=99%); |
/=9GB;207;219;0;231 /dev=0GB;0;0;0;0
```

Voilà, NRPE fonctionne correctement en local. Il ne reste plus qu'à vérifier que l'on arrive à le joindre depuis notre serveur Nagios.

Si comme moi vous avez obtenu ce message dans vos log :

```
/var/log/messages
```

```
mpt-statusd: detected non-optimal RAID status
```



En effet, lorsque Debian s'installe sur un ESXi (machine virtuelle), il considère que les disques en dessous sont en RAID et installe les modules de vérification du RAID. Afin que cela cesse, il faut simplement désinstaller le paquet suivant :

```
apt-get remove mpt-status
```

Si vous rencontrez ce message, ne soyez pas surpris.

```
connect to address ::1 port 5666: Connection refused
```

« ::1 » est l'adresse loopback IPv6, l'équivalent de celui en IPv4 127.0.0.1. Et comme vous le savez, la version 2.15 de **NRPE supporte maintenant IPV6**.

III-B-2-b - Installation de NRPE sur le serveur Nagios

Ce sont les mêmes commandes que pour le **client Linux** :



Si vous faites une mise à jour, il est important de faire une sauvegarde du fichier /usr/local/nagios/etc/nrpe.cfg !

```
supervision:~ # cd /usr/local/src/
supervision:/usr/local/src # wget https://sourceforge.net/projects/nagios/files/nrpe-3.x/nrpe-3.1.1/nrpe-3.1.1.tar.gz
```

```

supervision:/usr/local/src # tar -xzf nrpe-3.1.1.tar.gz
supervision:/usr/local/src # cd nrpe-3.1.1
supervision:/usr/local/src/nrpe-3.1.1 # ./configure --enable-command-args
supervision:/usr/local/src/nrpe-3.1.1 # make all
supervision:/usr/local/src/nrpe-3.1.1 # make install
supervision:/usr/local/src/nrpe-3.1.1 # make install-config
supervision:/usr/local/src/nrpe-3.1.1 # make install-init
supervision:/usr/local/src/nrpe-3.1.1 # systemctl enable nrpe.service
    
```



Pour tout souci, veuillez vous reporter au [chapitre précédent](#).

Maintenant, testons des commandes NRPE distantes

```

supervision:~ # /usr/local/nagios/libexec/check_nrpe -H servprod -c check_disk -a 50% 100% /
DISK CRITICAL - free space: / 227596 MB (96% inode=-); | /=9371MB;118483;0;0;236967
    
```

Essayons maintenant de l'intégrer à Nagios. Il faut définir un objet command et les services dans servprod.cfg. Commençons par créer un répertoire serveurs_linux où déposer notre fichier servprod.cfg.

```

mkdir /usr/local/nagios/etc/serveurs_linux/
    
```

Rajoutons le chemin de ce répertoire dans nagios.cfg.

```

vi /usr/local/nagios/etc/nagios.cfg
...
cfg_dir=/usr/local/nagios/etc/serveurs_windows
cfg_dir=/usr/local/nagios/etc/serveurs_linux
...
    
```

Ouvrons le fichier « commandes.cfg »

```

vi /usr/local/nagios/etc/objects/commands.cfg
    
```

et rajoutons l'objet commande pour nos requêtes NRPE.

```

# 'check_nrep' command definition
define command{
    command_name    check_nrpe
    command_line    $USER1$/check_nrpe -H $HOSTADDRESS$ -u -c $ARG1$ -
a $ARG2$ $ARG3$ $ARG4$ $ARG5$ $ARG6$
}
    
```

Maintenant, créons un fichier « groupe_serveurs_linux.cfg » dans lequel on définira notre groupe de serveurs Linux. Ainsi, à chaque rajout de serveur Linux, on fera la modification dans ce fichier.

```

$ vi /usr/local/nagios/etc/serveurs_linux/groupe_serveurs_linux.cfg
# Define a hostgroup for Linux machines
# All hosts that use the linux-server template will automatically be a member of this group

define hostgroup{
    hostgroup_name linux-servers ; The name of the hostgroup
    alias          Serveurs Linux ; Long name of the group
    members        servprod      ; separees par des virgules
}
    
```

La définition du hostgroup « linux-servers » pour les Linux se trouve dans le fichier /usr/local/nagios/etc/objects/localhost.cfg (fichier de configuration du serveur supervision). Mettons en commentaire cette définition :

```

vi /usr/local/nagios/etc/objects/localhost.cfg
#####
#####
#
# HOST GROUP DEFINITION
#
#####
#####

# Define an optional hostgroup for Linux machines

#define hostgroup{
#     hostgroup_name  linux-servers ; The name of the hostgroup
#     alias           Linux Servers ; Long name of the group
#     members         localhost      ; Comma separated list of hosts that belong to this group
#     }
    
```

Créons notre fichier « servprod.cfg ».

```
vi /usr/local/nagios/etc/serveurs_linux/servprod.cfg
```

Voici un exemple de fichier servprod.cfg

servprod.cfg

```

# definition de la machine

define host{
    use          linux-server ;
    host_name    servprod      ;
    alias        Serveur servprod ;
    address      servprod      ;
}

# Verification de l'espace disque

define service{
    use          generic-service
    host_name    servprod
    service_description  Espace disque /
    check_command    check_nrpe!check_disk!80%!90%!/
}

# Verification de la charge cpu

define service{
    use          generic-service
    host_name    servprod
    service_description  Charge CPU
    check_command    check_nrpe!check_load!80!90
}

# Verification du nombre d'utilisateurs sur le serveur

define service{
    use          generic-service
    host_name    servprod
    service_description  Nombre utilisateurs
    check_command    check_nrpe!check_users!2!10
}
    
```



Pensez à attribuer les bons droits Nagios sur les fichiers de configuration :

Droits

```
chown -R nagios.nagios /usr/local/nagios/etc/serveurs_*
```

Redémarrage de Nagios.

```
# /usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg
# /etc/init.d/nagios stop; pkill nagios; /etc/init.d/nagios start
```

III-B-3 - Switch à surveiller

À ce stade, je pense que vous commencez déjà à comprendre le fonctionnement de Nagios. Il existe un fichier d'exemple de configuration pour les switches (/usr/local/nagios/etc/objects/switch.cfg), créons un répertoire /usr/local/nagios/etc/switches, rajoutons ce fichier à l'intérieur.

```
supervision:~ # cp /usr/local/nagios/etc/objects/switch.cfg /usr/local/nagios/etc/switches/
switch3com
```

Dans nagios.cfg, par défaut la ligne définissant le chemin où trouver les fichiers de configuration des switches existe déjà, mais elle est commentée, il suffit donc de la décommenter.

/usr/local/nagios/etc/nagios.cfg

```
# directive as shown below:

cfg_dir=/usr/local/nagios/etc/serveurs_windows
cfg_dir=/usr/local/nagios/etc/serveurs_linux
#cfg_dir=/usr/local/nagios/etc/servers
#cfg_dir=/usr/local/nagios/etc/printers
cfg_dir=/usr/local/nagios/etc/switches
#cfg_dir=/usr/local/nagios/etc/routers
```

Maintenant, modifions notre fichier /usr/local/nagios/etc/switches/switch3com.cfg en précisant l'adresse IP du switch. Enlevez la section hostgroup pour le mettre dans un fichier à part. Nos fichiers ressembleront à :

/usr/local/nagios/etc/switches/switch3com.cfg

```
#####
# SWITCH.CFG - SAMPLE CONFIG FILE FOR MONITORING A SWITCH
#
# Last Modified: 10-03-2007
#
# NOTES: This config file assumes that you are using the sample configuration
#       files that get installed with the Nagios quickstart guide.
#
#####

#####
#####
#
# HOST DEFINITIONS
#
#####
#####

# Define the switch that we'll be monitoring

define host{
    use         generic-switch           ; Inherit default values from a template
    host_name   MONSWITCH3COM           ; The name we're giving to this switch
    alias       Linksys SRW224P Switch   ; A longer name associated with the switch
    address     192.168.1.1              ; IP address of the switch
}
```

/usr/local/nagios/etc/switches/switch3com.cfg

```

hostgroups switches ; Host groups this switch is associated with
}

#####
#####
#
# SERVICE DEFINITIONS
#
#####
#####

# Create a service to PING to switch

define service{
    use generic-service ; Inherit values from a template
    host_name MONSWITCH3COM ; The name of the host the service is associated with
    service_description PING ; The service description
    check_command check_ping!200.0,20%!600.0,60% ; The command used to monitor the
    service
    normal_check_interval 5 ; Check the service every 5 minutes under normal conditions
    retry_check_interval 1 ; Re-check the service every minute until its final/hard
    state is determined
}

# Monitor uptime via SNMP

define service{
    use generic-service ; Inherit values from a template
    host_name MONSWITCH3COM
    service_description Uptime
    check_command check_snmp!-C public -o sysUpTime.0
}

# Monitor Port 1 status via SNMP

define service{
    use generic-service ; Inherit values from a template
    host_name MONSWITCH3COM
    service_description Port 1 Link Status
    check_command check_snmp!-C public -o ifOperStatus.1 -r 1 -m RFC1213-MIB
}

# Monitor bandwidth via MRTG logs

define service{
    use generic-service ; Inherit values from a template
    host_name MONSWITCH3COM
    service_description Port 1 Bandwidth Usage
    check_command check_local_mrtgtraf!/var/lib/mrtg/192.168.1.253_1.log!
    AVG!1000000,1000000!5000000,5000000!10
}
    
```

/usr/local/nagios/etc/switches/hostgroup_switch.cfg

```

# HOST GROUP DEFINITIONS
#

# Create a new hostgroup for switches

define hostgroup{
    hostgroup_name switches ; The name of the hostgroup
    alias Les Switches ; Long name of the group
}
    
```

Il ne reste plus qu'à redémarrer Nagios.

```
# /usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg
# /etc/init.d/nagios stop; pkill nagios; /etc/init.d/nagios start
```

Dernière petite explication d'un service de switch. Ce service lance une commande SNMP avec pour arguments -C public -o ifOperStatus.1 -r 1 -m RFC1213-MIB :

```
# Monitor Port 1 status via SNMP

define service{
    use                generic-service      ; Inherit values from a template
    host_name          MONSWITCH3COM
    service_description Port 1 Link Status
    check_command       check_snmp!-C public -o ifOperStatus.1 -r 1 -m RFC1213-MIB
}
```

- Explication issue de la documentation officielle

-o ifOperStatus.1 fait référence à OID pour l'état opérationnel du port 1 sur le switch.

L'option -r 1 indique au plug-in check_snmp de retourner un état OK si 1 est trouvé dans la réponse SNMP (1 indique que le port est up) et CRITICAL sinon. L'option -m RFC1213-MIB est facultative et indique la MIB à utiliser parmi celles installées sur votre système et peut aider à accélérer les choses. Voilà pour l'exemple de supervision avec SNMP. Il y a des millions de choses qui peuvent être supervisées par SNMP, aussi est-ce à vous de décider ce dont vous avez besoin et ce que vous souhaitez superviser. Bonne chance !

ASTUCE

Vous pouvez trouver les OID qui peuvent être supervisés sur un switch en exécutant la commande suivante :

```
snmpwalk -v1 -c public IP_DE_VOTRE_SWITCH -m ALL .1
```

IV - Mise à jour de Nagios 2.x/3.x vers 3.4.1, 4.x vers 4.x

Il est important de s'informer des nouvelles mises à jour de Nagios. Il est donc important de connaître la démarche à suivre, car il est inutile de devoir tout réinstaller. Je vais vous exposer les lignes de commande à lancer.

N.B. Je me suis basé sur la documentation officielle de Nagios, n'hésitez pas à y jeter un coup d'œil pour de plus amples informations. Pour savoir si votre Nagios est à jour, il suffit de regarder la page d'accueil Web de votre Nagios.

Nagios propose trois types de mises à jour :

- mise à jour depuis une version 3.x de Nagios ;
- mise à jour depuis une version 2.x de Nagios ;
- mise à jour depuis une installation RPM.

Dans cette documentation, nous allons exposer la première méthode, plus précisément en passant de Nagios 3.3.1 à Nagios 3.4.1 sur notre serveur nommé « supervision ».

La mise à jour ne nécessite pas les droits root. Il faut se connecter avec l'utilisateur « nagios ».

Sous Debian ou Ubuntu

```
sudo -s nagios
```

Sous les autres OS Linux

```
supervision:~ # su -l nagios
nagios@supervision:~>
```

Afin de rester dans la même logique de notre article, plaçons-nous dans le répertoire où nous avons téléchargé les fichiers compressés de Nagios (sinon créez ce répertoire).

```
nagios@supervision:~> cd /nagios/download
nagios@supervision:/nagios/download>
```

Téléchargeons maintenant la dernière version de Nagios 3.4.1 (à ce jour 29/08/2012).

Téléchargement

```
nagios@supervision:/nagios/download> wget http://prdownloads.sourceforge.net/sourceforge/nagios/nagios-3.4.1.tar.gz
```

Nous pouvons extraire le code source de l'archive.

```
nagios@supervision:/nagios/download>tar -xvzf nagios-3.4.1.tar.gz
```

On pourrait s'attendre à la création d'un répertoire nagios-3.4.1, mais on a plutôt un répertoire nagios.

```
nagios@supervision:/nagios/download> ls -alh
drwxr-xr-x  5 nagios nagios 4,0K oct. 26 11:56 .
drwxr-xr-x  3 root  root  4,0K janv. 18 2011 ..
drwxr-xr-x 16 nagios users 4,0K juil. 25 23:20 nagios
-rw-r--r--  1 nagios nagios 1,7M juil. 26 05:32 nagios-3.4.1.tar.gz
nagios@supervision:/nagios/download> cd nagios
nagios@supervision:/nagios/download/nagios>
```

Lançons le programme de configuration de Nagios.

```
nagios@supervision:/nagios/download/nagios> ./configure --with-command-group=nagcmd
```

Compilons le code source de Nagios et lançons l'installation de la mise à jour des binaires. Cette étape n'engendre pas l'écrasement de vos fichiers de configuration.

```
nagios@supervision:/nagios/download/nagios> make all && make install
```

Nous pouvons maintenant tester notre configuration de Nagios et le redémarrer.

```
nagios@supervision:/nagios/download/nagios> /usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg
```

Ensuite, relançons Nagios (en root).

```
supervision:/nagios/download # /etc/init.d/nagios restart
```

Si tout s'est bien déroulé, votre Nagios est à jour. Vous le constaterez en allant sur la page d'accueil de l'interface Nagios.



N'ayant pas testé une mise à jour de 3.x vers 4.x, je n'en parle pas afin d'éviter tout problème. En ce qui me concerne, j'avais réinstallé un nouveau serveur, donc je suis parti directement sur une 4.x, d'où la version utilisée dans le tutoriel.



J'ai effectué les commandes ci-dessus pour migrer un serveur Nagios version 4.0.1 vers la 4.0.7 sans souci.

V - Divers

V-A - Installation openssl, openssl-devel

- Installation sous « Suse LinuxEnterprise Server »

L'installation « openssl-devel » n'est pas toujours évidente, car il n'est pas disponible dans la distribution par défaut et ce n'est pas simple de le trouver dans les dépôts. Voici un lien qui m'a déjà dépanné et qui contient quelques RPM : http://www.convirture.com/repos/deps/SLES/11.x/x86_64/. Il contient des RPM pour la distribution « SLES 11.x ». Si le lien n'est plus fonctionnel, vous pouvez les trouver ici :

- 1 [libopenssl-devel-0.9.8h-30.11.x86_64.rpm](#) ;
- 2 [libopenssl-devel-0.9.8j-0.24.1.x86_64.rpm](#) ;
- 3 [libopenssl-devel-0.9.8j-0.26.1.x86_64.rpm](#) ;
- 4 [zlib-devel-1.2.3-106.34.x86_64.rpm](#).

Pour l'installation, voici comment je m'y suis pris.

Ma version de libssl

```
# find / -name libssl.so*
/usr/lib64/libssl.so.0.9.8
/usr/lib/libssl.so.0.9.8
```

Je vais dans « Yast » pour trouver la description exacte d'openssl sur le serveur. Je constate que je dispose de « libopenssl0_9_8 : 0.9.8j-0.26.1 (0.9.8j-0.50.1) ». J'ai donc installé « libopenssl-devel-0.9.8j-0.26.1.x86_64.rpm ».

Installation libopenssl-devel

```
# cd /tmp
# wget http://djibril.developpez.com/tutoriels/linux/nagios-pour-debutant/fichiers/zlib-
devel-1.2.3-106.34.x86_64.rpm
# wget http://djibril.developpez.com/tutoriels/linux/nagios-pour-debutant/fichiers/libopenssl-
devel-0.9.8j-0.26.1.x86_64.rpm
# rpm -ivh zlib-devel-1.2.3-106.34.x86_64.rpm
# rpm -ivh libopenssl-devel-0.9.8j-0.26.1.x86_64.rpm
# find / -name libssl.so*
/usr/lib64/libssl.so
/usr/lib64/libssl.so.0.9.8
/usr/lib/libssl.so.0.9.8
```

V-B - Erreurs

V-B-1 - Could not complete SSL handshake

J'ai déjà eu cette erreur pour un de mes serveurs alors que tout était OK quelques jours avant. Par exemple, depuis mon serveur Nagios, en ligne de commande, voici ce que j'obtenais :

```
# /usr/local/nagios/libexec/check_nrpe -H serveur distant
CHECK_NRPE: Error - Could not complete SSL handshake.
```

La première chose à faire est de vérifier sur le serveur surveillé son fichier « /etc/xinetd.d/nrpe » afin de vérifier que la ligne `only_from` contient bien l'adresse IP ou le nom DNS sur serveur Nagios.

```
# cat /etc/xinetd.d/nrpe
# default: on
# description: NRPE (Nagios Remote Plugin Executor)
service nrpe
{
```

```

    flags          = REUSE
    socket_type    = stream
    port           = 5666
    wait           = no
    user           = nagios
    group          = nagios
    server         = /usr/local/nagios/bin/nrpe
    server_args    = -c /usr/local/nagios/etc/nrpe.cfg --inetd
    log_on_failure += USERID
    disable        = no
    only_from      = 127.0.0.1 SERVEURNAGIOS.domain
}

```

Ensuite, il est très important de vérifier que le DNS de son réseau fonctionne correctement. Dans mon cas, c'était le souci. Ce fichier était bien configuré, mais j'avais un souci réseau, un souci DNS, mon fichier `/etc/resolv.conf` contenait une IP erronée. Je l'ai corrigé, et j'ai relancé les services `network` et `xinetd` et tout est rentré dans l'ordre.

```

serveur #/etc/init.d/network restart
serveur #/etc/init.d/xinetd restart

gendarme:~# /usr/local/nagios/libexec/check_nrpe -H serveur.domaine
NRPE v2.15

```

V-B-2 - Page URL Could not complete SSL handshake

J'ai déjà eu cette erreur pour un de mes serveurs alors que tout était OK quelques jours avant. Par exemple, depuis mon serveur Nagios, en ligne de commande, voici ce que j'obtenais :

```

# /usr/local/nagios/libexec/check_nrpe -H serveur distant
CHECK_NRPE: Error - Could not complete SSL handshake.

```

La première chose à faire est de vérifier sur le serveur surveillé son fichier « `/etc/xinetd.d/nrpe` » afin de vérifier que la ligne `only_from` contient bien l'adresse IP ou le nom DNS sur serveur Nagios.

```

# cat /etc/xinetd.d/nrpe
# default: on
# description: NRPE (Nagios Remote Plugin Executor)
service nrpe
{
    flags          = REUSE
    socket_type    = stream
    port           = 5666
    wait           = no
    user           = nagios
    group          = nagios
    server         = /usr/local/nagios/bin/nrpe
    server_args    = -c /usr/local/nagios/etc/nrpe.cfg --inetd
    log_on_failure += USERID
    disable        = no
    only_from      = 127.0.0.1 SERVEURNAGIOS.domain
}

```

Ensuite, il est très important de vérifier que le DNS de son réseau fonctionne correctement. Dans mon cas, c'était le souci. Ce fichier était bien configuré, mais j'avais un souci réseau, un souci DNS, mon fichier `/etc/resolv.conf` contenait une IP erronée. Je l'ai corrigé, et j'ai relancé les services `network` et `xinetd` et tout est rentré dans l'ordre.

```

serveur #/etc/init.d/network restart
serveur #/etc/init.d/xinetd restart

gendarme:~# /usr/local/nagios/libexec/check_nrpe -H serveur.domaine
NRPE v2.15

```

V-B-3 - Page supervision not found

À la fin de votre installation, vous êtes heureux de pouvoir voir votre page Web de supervision : <http://supervision/nagios/>, et là c'est le drame 😞 !

Pas de panique, vous avez un souci au niveau d'Apache qui durant son démarrage n'a pas chargé la configuration de Nagios. Il se peut que la commande suivante `make install-webconf` au moment de l'installation de Nagios n'ait pas correctement fonctionné.

```
root@supervision:/nagios/download/nagios-4.3.2# make install-webconf
/usr/bin/install -c -m 644 sample-config/httpd.conf /etc/apache2/conf.d/nagios.conf
if [ 0 -eq 1 ]; then \
    ln -s /etc/apache2/conf.d/nagios.conf /etc/apache2/sites-enabled/nagios.conf; \
fi
*** Nagios/Apache conf file installed ***
```

Vérifiez la présence du fichier `/etc/apache2/conf.d/nagios.conf` sur votre serveur qui contient la configuration du site virtuel de Nagios. Lancez en ligne de commande la création du lien symbolique :

```
root@supervision:/nagios/download/nagios-4.3.2# ln -s /etc/apache2/conf.d/nagios.conf /etc/
apache2/sites-enabled/nagios.conf
root@supervision:/nagios/download/nagios-4.3.2# ls -alh /etc/apache2/sites-enabled/
total 8,0K
drwxr-xr-x 2 root root 4,0K juin  9 15:15 .
drwxr-xr-x 9 root root 4,0K juin  9 15:13 ..
lrwxrwxrwx 1 root root  35 juin  8 17:15 000-default.conf -> ../sites-available/000-
default.conf
lrwxrwxrwx 1 root root  31 juin  9 15:15 nagios.conf -> /etc/apache2/conf.d/nagios.conf
```

On constate que l'on a un lien symbolique supplémentaire dans le répertoire des sites d'Apache. Redémarrez Apache !

```
# /etc/init.d/apache2 restart
```

Vous devriez avoir la possibilité de vous connecter maintenant à <http://supervision/nagios/>. Si ce n'est pas le cas, il vous manque peut-être des prérequis à installer (voir [chapitre 1](#)).

V-B-4 - Could not connect to xxx.xxx.xxx.xxx : Connection reset by peer

Sur un de mes serveurs Debian, après une mise à jour de `nrpe` 1.5 à 3.1.1, la supervision à distance ne fonctionnait plus.

```
~# /usr/local/nagios/libexec/check_nrpe -H servprod
CHECK_NRPE: Error - Could not connect to 192.168.200.xx: Connection reset by peer
```

Sur le `servprod` en local tout était OK, mais dans les log, j'avais plus d'informations sur les erreurs d'administration à distance :

```
# /usr/local/nagios/libexec/check_nrpe -H 127.0.0.1
NRPE v3.1.1
# tail -f /var/log/message
nrpe[1957]: Host 192.168.200.xx is not allowed to talk to us!
nrpe[1957]: Host 192.168.200.xx is not allowed to talk to us!
nrpe[1957]: Host 192.168.200.xx is not allowed to talk to us!
```

Tout était pourtant bien installé (`nrpe`, les prérequis...). J'ai finalement trouvé la solution dans la documentation officielle au niveau de la section mise à jour que vous expose ci-dessous.

Les précédentes versions NRPE utilisaient « inetd » ou « xinetd » pour démarrer ou arrêter le daemon NRPE. L'installation de la nouvelle version 3 crée son propre service et n'utilise plus inetd/xinetd.

Pour installer le service/démon, on peut soit ne plus utiliser le service inetd/xinetd, soit l'utiliser. J'ai donc décidé de ne plus l'utiliser de la façon suivante :

Arrêt du service

```
/etc/init.d/xinetd stop
```

Ouverture du fichier `/etc/xinetd.d/nrpe` et copie du contenu de la ligne `only_from`.

```
# cat /etc/xinetd.d/nrpe
# default: on
# description: NRPE (Nagios Remote Plugin Executor)
service nrpe
{
    flags          = REUSE
    socket_type    = stream
    port          = 5666
    wait          = no
    user          = nagios
    group         = nagios
    server        = /usr/local/nagios/bin/nrpe
    server_args   = -c /usr/local/nagios/etc/nrpe.cfg --inetd
    log_on_failure += USERID
    disable       = no
    only_from     = 127.0.0.1 SERVEURNAGIOS.domain
}
```

Puis, supprimer le fichier `/etc/xinetd.d/nrpe`.

Maintenant, ouvrez le fichier `/usr/local/nagios/etc/nrpe.cfg`. Au niveau de la ligne `allowed_hosts=` , rajoutez `SERVEURNAGIOS.domain`...

```
allowed_host=127.0.0.1, SERVEURNAGIOS.domain
```

Sauvegarde du fichier et installation des services et démons notamment sous Debian :

```
root@servprod:/usr/local/src/nrpe-3.1.1# systemctl enable nrpe.service
```

Redémarrage du serveur et tout est OK.

VI - Liens

Liste des sites ayant servi à la réalisation de cet article

- [🇬🇧 Site de Nagios.](#)
- [🇬🇧 Les plug-ins Nagios \(documentation officielle\).](#)
- [🇬🇧 NSClient++.](#)
- [🇫🇷 Traduction de la documentation officielle de Nagios.](#)
- [🇬🇧 Nagios Support Knowledgebase.](#)
- [🇫🇷 Concevoir facilement un plug-in Nagios en Perl.](#)

VII - Conclusion

J'espère que cet article vous a aidé à débiter et installer Nagios sur votre serveur. N'hésitez pas à faire des remarques, corrections ou appréciations : **Commentez**

VIII - Remerciements

Je remercie l'équipe developpez.com pour les remarques et la relecture de cet article, notamment **ClaudeLELOUP**, **stoyak** et **Torgar**.