Cahier de supervision

H4413

11 mai 2011

Table des matières

Première partie
Monitoring depuis un poste Windows

1 Introduction

Un serveur de supervision (SNMP/MRTG et/ou Nagios/NRPE) peut être configuré puis utilisé sans problèmes depuis une plateforme Windows ; il suffit pour cela de disposer d'un accès ssh vers la machine exécutant les logiciels de supervision, puis d'utiliser son éditeur de texte préféré pour configurer à volonté le serveur.

Pour interfacer Nagios et Windows, un programme tiers est nécessaire ; on propose ici d'utiliser NSClient++ dans sa version 0.2.5e.

2 Installation de NSClient++

L'installation se fait à partir d'une invite de commande MS-Dos. Dans le dossier où l'archive NSClient++ a été décompressée, lancer la commande NSClient++ /install.

3 Configuration de NSClient++

Plusieurs choses doivent être faites :

3.1 Permettre à NSClient++ d'interagir avec le bureau

Il sera ainsi en mesure d'envoyer vers le serveur Nagios des informations concernant l'ordinateur monitoré. On procède de la façon suivante : Menu Démarrer/Accessoires/Outils d'administration/Service ; une fois ici il faut trouver le service associé à NSClient et lui donner l'autorisation.

3.2 Configuration de NSClient++ proprement dite

Elle s'effectue en éditant le fichier nsc.ini situé dans le répertoire d'installation du client. On y

1. décommente les modules que l'on souhaite utiliser
2. redéfinit le mot de passe
3. ajout le serveur Nagios dans la section Allowed Hosts (les différentes adresses sont ici séparées par des virgules)
4. définit le port associé à NSClient++ en décommentant la ligne correspondante

Après tout cela il faut relancer le service. On le fait grâce aux commandes suivantes :

```
NSClient++ /stop
NSClient++ /start
```

3.3 Contrôles

Pour s'assurer que l'installation fonctionne, il faut tout d'abord vérifier que le port choisi à la section précédente est bien ouvert dans les différents pare-feux protégeant l'ordinateur.
Une autre vérification est à mener au niveau du serveur Nagios. La commande suivante :

```
cd /usr/local/nagios/libexec # a adapter suivant votre installation
./check_nt -H xxx.xxx.xxx.xxx -v CLIENTVERSION -p 12489
```

4 enregistrement du poste windows

Il suffit pour cela d'éditer le fichier nagios.cfg en décommentant la ligne correspondant à windows.cfg, puis d'éditer windows.cfg avec le nom et l'adresse IP de la machine à monitorer. Ne pas oublier de redémarrer Nagios pour que les changements soient pris en compte !

Deuxième partie

Monitoring d'un serveur Linux avec MRTG

5 Introduction

Mrtg (Multi Router Traffic Grapher) est un outil graphique fournissant les statistiques d'un serveur. Vous allez pouvoir connaître la RAM disponible, la charge CPU, la taille du SWAP, le trafic réseau, etc.

Il génère un graphique journalier, hebdomadaire, mensuel et annuel, ce qui permet une bonne vue d'ensemble. Il est ainsi très aisé de détecter une saturation du serveur, que se soit au niveau du trafic réseau que de la charge du processeur.

6 Installation

Sous une distribuition basé en Debian, l'installation est rapide et efficace :

apt-get install snmp mrtg snmpd

SNMP signifie Simple Network Management Protocol, il permet de vérifier que le réseau fonctionne bien et peut retourner des informations systèmes telles que la charge du processeur, ce qui est particulièrement intéressant pour nous.

MRTG va nous générer les graphiques à partir des données récupérées par SNMP.

7 Configuration SNMP

On va commencer par éditer le fichier de configuration SNMP

vim /etc/snmp/snmpd.conf

Il faut autoriser l'accès en lecture des données SNMP.

Par défaut, la ligne est décommentée. Entre la partie "First," et "Second" du fichier de configuration, il faut commenter la ligne "com2sec paranoid default public" en rajoutant un "#" puis supprimer le "#" de la ligne "com2sec readonly default public".

```
####
# First, map the community name (COMMUNITY) into a security name
# (local and mynetwork, depending on where the request is coming
# from):

# sec.name source community
#com2sec paranoid default public
com2sec readonly default public
#com2sec readwrite default private

####
# Second, map the security names into group names:
```

Modifier la partie "syslocation" en mettant le pays où se trouve votre serveur, donc "syslocation France", il faut également modifier le contact "syscontact", j'ai mis "syscontact MonPseudo <root@localhost>"

```
################################
# System contact information
#

# It is also possible to set the sysContact and sysLocation system
# variables through the snmpd.conf file. **PLEASE NOTE** that setting
# the value of these objects here makes these objects READ-ONLY
# (regardless of any access control settings). Any attempt to set the
# value of an object whose value is given here will fail with an error
```

```
# status of notWritable.

syslocation France (configure /etc/snmp/snmpd.local.conf)
syscontact Zigzig <root@localhost> (configure /etc/snmp/snmpd.local.conf)
```

Comme toutes les modifications, il faut redémarrer le service, on redémarre donc SNMP :

```
/etc/init.d/snmpd restart
```

8 Configuration MRTG

Nous allons créer le dossier qui va contenir les graphiques :

```
mkdir /var/www/blog/stats/mrtg
```

Bien évidemment, vous pouvez spécifier un autre dossier ;)
Nous allons générer un début de configuration pour MRTG :

```
cfgmaker
--global 'WorkDir: /var/www/blog/stats/mrtg'
--global 'Language: french'
--global 'Options[_]: bits,growright'
--ifdesc=descr public@localhost
--output /etc/mrtg.cfg
```

"WorkDir" désigne l'emplacement où seront enregistrés les graphiques. Les miens sont stockés dans le dossier "/var/www/blog/stats/mrtg" "Language : french" désigne la langue, on met donc "french" pour Français. "Options[_] : bits,growright" on définit l'unité de mesure en bits. "–output /etc/mrtg.cfg" désigne l'emplacement du fichier de configuration de mrtg : "/etc/mrtg.cfg". Ainsi, le fichier sera généré dans le repertoire "/etc" et aura comme nom "mrtg.cfg"
Nous pouvons maintenant voir le fichier de configuration :

```
vim /etc/mrtg.cfg
```

Nous le modifierons plus tard.
Nous allons générer la page HTML pour voir ce que MRTG peut nous faire :

```
indexmaker /etc/mrtg.cfg --output=/var/www/blog/stats/mrtg/index.html
```

"indexmaker" est l'outil pour générer les pages html de MRTG. "/etc/mrtg.cfg" lui indique le fichier de configuration "–output=/var/www/blog/stats/mrtg/index.html" définit l'endroit ou sera stocké les pages HTML.
Les pages sont crées. Pour mettre les graphiques à jour, exécutez :

```
/usr/bin/mrtg /etc/mrtg.cfg
```

si le paquet est installé par apt-get install mrtg, le paquet configure Cron tout seul pour que les graphiques soient mises à jours toutes les 5 minutes donc vous n'avez besoin de suivre les 4 lignes suivantes :
Nous allons utiliser CronTab pour éviter de le mettre à jour à la main.

```
crontab -e
```

On rajoute un ligne :

```
0-59/5 * * * * /usr/bin/mrtg /etc/mrtg.cfg
```

Ce qui veut dire que la mise à jour sera exécutée par le système toutes les 5 minutes.

9 Plus de monitoring

Pour le moment, seul le trafic réseau est visible, nous allons rajouter le monitoring du CPU, de la SWAP, de la RAM.

On édite le fichier de configuration MRTG :

```
vim /etc/mrtg.cfg
```

On rajoute à la fin la charge CPU :

```
#---------CPU-----------------------
Target[cpu]:ssCpuRawUser.0&ssCpuRawUser.0:public@localhost + ssCpuRawSystem.0&ssCpuRawSystem.0:pub
RouterUptime[cpu]: public@localhost
MaxBytes[cpu]: 100
Title[cpu]: CHARGE CPU
PageTop[cpu]: Charge Active CPU \%
Unscaled[cpu]: ymwd
ShortLegend[cpu]: \%
YLegend[cpu]: Utilisation CPU
Legend1[cpu]: CPU Actif en \% (Charge)
Legend2[cpu]:
Legend3[cpu]:
Legend4[cpu]:
LegendI[cpu]: Actif
LegendO[cpu]:
Options[cpu]: growright,nopercent
#--------end CPU-----------------------
```

Ainsi, on additionne la charge utilisateur et système pour avoir la charge totale.

Pour la RAM et le SWAP, on rajoute ce code à la suite de celui de la charge du processeur :

```
#---------SWAP--------------------
Target[swap]: memAvailSwap.0&memTotalSwap.0:public@localhost
Options[swap]: nopercent,growright,gauge,noinfo
Title[swap]: Swap
PageTop[swap]: Swap
MaxBytes[swap]: 1000000000
kMG[swap]: k,M,G,T,P,X
Ylegend[swap]: Octets
ShortLegend[swap]: octets
LegendI[swap]: Swap dispo
LegendO[swap]: Swap total
Legend1[swap]: Swap disponible
Legend2[swap]: Swap total
#--------end SWAP----------------------
```

```
#---------RAM-------------------
Target[ram]: memAvailReal.0&memTotalReal.0:public@localhost
Options[ram]: nopercent,growright,gauge,noinfo
Title[ram]: RAM
PageTop[ram]: RAM.
MaxBytes[ram]: 1000000000
kMG[ram]: k,M,G,T,P,X
Ylegend[ram]: Octets
ShortLegend[ram]: octets
LegendI[ram]: RAM dispo
LegendO[ram]: RAM total
Legend1[ram]: RAM disponible
Legend2[ram]: RAM total
#--------end RAM--------------------
```

Explication :

MRTG utilise 2 courbes, il faut donc 2 données. On lui indique donc 2 sources de données : "memAvailReal.0" retourne la taille de la mémoire vive disponible, et "memTotalReal" retourne la taille de ma mémoire vive totale.

On fait la même chose pour le SWAP.

10 Conclusion

MRTG permet de générer des graphiques simples de façon à détecter des dysfonctionnements ou des surcharges.

Ainsi, un graphique CPU à 100% prouve que le processeur n'est pas assez performant, ou bien qu'un script ne fonctionne pas correctement et effectue une boucle sans fin.

Troisième partie

Monitoring d'un serveur Linux avec Nagios/NRPE

11 Installation Nagios

Sous Fedora :

```
# sudo yum install nagios3
```

Pour installer un serveur Nagios sous Ubuntu/Debian, il suffit :

```
# sudo apt-get install nagios3
```

12 Lancer Nagios

Vérifier qu'il n'a pas d'erreurs dans l'installation de Nagios :

```
# /usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg
```

On devrait retrouver quelque chose comme ça :

```
Total Warnings: 0
Total Errors:   0
Things look okay - No serious problems were detected during the pre-flight check
```

Donc pour lancer Nagios, il suffit d'écrire :

```
# service nagios start
```

13 Installation NRPE

Il faut installer le plugin NRPE. Pour cela, le plus simple est de faire confiance à votre gestionnaire de paquets. Sous Fedora, la commande suivante devrait suffire :

```
# sudo yum install nagios-plugins-nrpe
```

Sous Ubuntu/Debian :

```
# sudo apt-get install nagios-nrpe-plugin
```

Il faut également vérifier que la définition du plugin est bien présente dans le fichier de configuration des commandes (commands.cfg) :

```
...
######
# NRPE
######
# 'check_nrpe' command definition
define command{
command_name check_nrpe
command_line \$USER1\$/check_nrpe -H \$HOSTADDRESS\$ -c \$ARG1\$
}
...
```

14 Serveur Linux à surveiller

La procédure est un peu plus longue. Il faut d'abord installer le daemon NRPE et les plugins Nagios (qui vont être lancés localement par le daemon NRPE) : Sous Ubuntu/Debian :

```
# sudo apt-get install nagios-nrpe-server
# sudo apt-get install nagios-plugins
```

Puis éditer le fichier /etc/nagios/nrpe.cfg pour modifier la ligne suivante :

```
...
allowed_hosts = Mettre ici l'adresse IP de votre serveur Nagios

...
```

On automatise le lancement du daemon au démarrage du serveur avec la commande :

```
# chkconfig --add nrpe
```

On ajoute une règle pour autoriser le Firewall IPtable à laisser passer les requêtes NRPE (à adapter selon vos règles) :

```
# iptables -I RH-Firewall-1-INPUT 10 -p tcp --dport 5666 -j ACCEPT
```

Attention il faut mettre deux-(- -) avant l'option dport
Il ne reste plus qu'à lancer le daemon : Sous Fedora :

```
# service nrpe start
```

Sous Ubuntu/Debien :

```
# /etc/init.d/nagios-nrpe-server start
```

15 Tester la communication

Pour tester que la communication entre le serveur Nagios et le serveur à surveiller se passe bien, il suffit de se rendre dans le répertoire des plugins (/usr/lib/nagios/plugins) de Nagios et de tester le plugin NRPE :

```
# ./check_nrpe -H Adresse_IP_du_serveur_Linux
NRPE v2.7
```

Si tout est OK, cette commande devrait renvoyer la version du daemon NRPE.
Vous pouvez tester directement les plugins avec la commande suivante (exemple donnée pour un check de la charge) :

```
# ./check_nrpe -H Adresse_IP_du_serveur_Linux -c check_load
```

16 Configurer Nagios

La dernière étape consiste à modifier les fichiers de configuration de Nagios pour intégrer le monitoring du/des serveur Linux. Il faut dans un premier temps éditer votre fichier de configuration des hosts (hosts.cfg par défaut) et y ajouter votre machine Linux :

```
define host {
use generic-host
host_name linus
alias Ma machine Linux
address 192.168.0.7
}
```

Puis ajouter les services offerts par NRPE (dans le fichier services.cfg), quelques exemples :

```
# Charge CPU
define service{
use generic-service
host_name remotehost
service_description CPU Load
check_command check_nrpe!check_load
}

# Memoire
define service{
use generic-service
host_name remotehost
service_description Memory
check_command check_nrpe!check_mem
}
```

Pour ajouter des nouveaux plugins executable par NRPE, il faut éditer le fichier /etc/nagios/nrpe.cfg et ajouter une ligne par service :

```
...
command[check_disk]=/usr/lib/nagios/plugins/check_disk -w 20 -c 10 -p /dev/hda
...
```

Ne pas oublier de relancer le daemon quand on change le fichier de configuration (nrpe.cfg) :

```
# service nrpe restart
```