TP1 : Supervision réseaux avec Nagios

Ce TP est à réaliser en binôme. Chaque binôme dispose de deux machines Linux : une machine NMS (manager) et une machine à superviser (agent). Sur la machine agent on installe un serveur FTP qui sera supervisé par la machine NMS en utilisant le logiciel NAGIOS.

Objectif

L'objectif de ce TP est de se familiariser avec l'outil de supervision Nagios. Nagios est un logiciel libre qui comprend deux composants : un service chargé d'interroger périodiquement les composants à superviser (PCs, commutateurs, routeurs, …) et une interface web permettant de visualiser graphiquement les résultats collectés, via une interface web.

1. Installation de Nagios

Nagios n'est pas présent dans les dépôts Debian. On ne peut donc pas l'installer via la commande apt-get. Les fichiers sources de ce logiciel sont disponibles sur le site suivant :

http://prdownloads.sourceforge.net/sourceforge/nagios/nagios-4.1.1.tar.gz

Le script suivant effectue l'installation complète du logiciel et des plugins associés :

```
#!/ bin/bash
if [ ! $UID -eq 0 ]
then
        echo 'Le script doit etre lancé en root.'
        exit 1
fi

useradd nagios
groupadd nagcmd
usermod -a -G nagcmd nagios
usermod -a -G nagcmd www-data
TMP_DIR=$(mktemp -d)
cd $TMP_DIR

wget http://prdownloads.sourceforge.net/sourceforge/nagios/nagios-4.1.1.tar.gz
tar xvzf nagios-4.1.1.tar.gz

cd nagios -4.1.1
./configure --with-command-group=nagcmd
make all
make install
make install-init
make install-config
make install-commandmode
make install-webconf
```

```
htpasswd -c /usr/local/nagios/etc/htpasswd.users nagiosadmin
/etc/init.d/apache2 reload

cd $TMP_DIR
wget http ://www.nagiosplugins.org/download/nagios-plugins-2.1.1.tar.gz
tar xvzf nagios-plugins-2.1.1.tar.gz
cd nagios-plugins-2.1.1
./configure --with-nagios-user=nagios --with-nagios-group=nagios
make
make install
```

Questions

1. Étudier le script d'installation et détailler la procédure d'installation effectuée par ce script
2. Exécuter ce script sur la machine NMS
3. Démarrer le service Nagios
4. Ouvrir un navigateur web à l'url suivante : http://localhost/nagios.
   Quelles sont les machines qui sont actuellement supervisées par le service nagios ?
   Et quelles sont les informations sur cet hôte qui sont collectées et affichées ?

2. Configuration de Nagios

Ajout d'un hôte

5. Sur la machine agent, installer sans le démarrer, le service proftpd (un serveur ftp).
6. Nous souhaitons superviser ce serveur FTP par l'application Nagios. Pour cela il faut modifier la configuration du service Nagios. Le fichier principale de configuration du service est :

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

   Tous les autres fichiers de configuration de nagios se trouvent dans le répertoire /usr/local/nagios/etc. Pour caque service à superviser nous fournissons un fichier de configuration : /usr/local/nagios/etc/objects/<service>.cfg qui doit être déclaré dans le fichier de configuration principal (nagios.cfg) en y ajoutant une ligne :

   cfg_file=/usr/local/nagios/etc/objects/<service>.cfg.

   Créer un fichier pour la supervision du serveur ftp lancé sur la machine agent. Dans le fichier de configuration de supervision de ce service on décrit le service à superviser par les instructions suivantes :

```
define host {
use             linux-server
host_name   serveur-ftp
address       <adresse-ip-du-serveur-ftp>
}
```

Redémarrer le service de Nagios. Qu'observe-t-on dans l'interface web de supervision ?

7. Pour l'instant la machine serveur a été déclarée, mais le service nagios n'effectue aucune opération de supervision sur cette machine.
   Dans le fichier de configuration /usr/local/nagios/etc/objects/<service>.cfg ajouter les lignes suivantes afin de tester périodiquement la disponibilité du service FTP.

```
define service {
use                     local-service
host_name               serveur-ftp
service_description     FTP
check_command           check_ftp
check_interval          1
}
```

Redémarrer le service de Nagios. Qu'observe-t-on dans l'interface web de supervision ?

8. Sur la machine serveur : démarrer le service proftpd. Dans l'interface web, cliquer sur le lien services. Qu'observe-t-on ?

2.1. Mise en place de notifications

On souhaite maintenant que l'administrateur (vous) soit prévenu par e-mail en cas de problème (p.ex., si un service n'est plus accessible). Pour cela on installe et on démarre sur la machine Nagios le service de mail postfix.

Créer le fichier /usr/local/nagios/etc/objects/admins.cfg qui contiendra les lignes suivantes permettant de déclarer deux contacts et un groupe d'administrateurs contenant ces deux contacts.

```
define contact {
contact_name        <nom_etudiant_1>
use                 generic-contact
email               <mail_etudiant_1>
}

define contact {
contact_name        <nom_etudiant_2>
use                 generic-contact
email               <mail_etudiant_2>
}

define contactgroup {
contactgroup_name       admins
members                 <nom_etudiant_1>,<nom_etudiant_2>
}
```

Ajouter dans la déclaration du service FTP les lignes suivantes :

```
notifications_enabled        1
first_notification_delay     0
notification_interval        0
```

9. Sur la machine agent : arrêter le service proftpd. Qu'observe-t-on ?

3. Supervision avancée

Les tâches de supervision exécutées par le service Nagios sont effectuées grâce à des fichiers exécutables appelés plugins et se trouvant dans le répertoire **/usr/lib/nagios/plugins**.

À l'installation de Nagios, ce répertoire contient de nombreux plugins permettant de réaliser des tests usuels.

La commande **check_ftp** utilisée précédemment pour vérifier la disponibilité du service FTP en est un exemple. Quand l'administrateur ne trouve pas dans ce répertoire de script répondant à ses besoins, il doit en développer de nouveaux.

Un plugin Nagios doit toujours avoir un code de retour parmi les trois valeurs suivantes : 0 (OK), 1 (avertissement ou warning) ou 2 (critique). Ces codes correspondent aux couleurs affichées dans l'interface web (0 = vert, 1 = orange et 2 = rouge).

L'objectif de cet exercice est d'écrire un plugin qui testera la gigue d'une liaison (jitter en anglais).

10. Écrire un script **check_jitter** qui enverra 10 demandes d'écho avec ping pour mesurer la gigue (appelée mdev par ping). Il prendra trois arguments : l'adresse IP de la machine à contacter, et deux seuils de gigue (en millisecondes) : un seuil d'avertissement et un seuil critique. Il aura un code de retour de :
    — 0 si la gigue est inférieure ou égale aux deux seuils passés en argument;
    — 1 si la gigue est strictement supérieure au seuil d'avertissement mais inférieure ou égale au seuil critique;
    — ou 2 dans tous les autres cas.

    Le script affichera aussi un message sur une seule ligne détaillant le résultat observé, par exemple :
    OK (si code = 0),
    Attention, gigue = 5 ms (si code = 1),
    Critique, gigue = 100 ms (si code = 2).

    On considérera pour simplifier que les seuils sont des entiers et on arrondira la gigue fournie par ping à l'entier inférieur.

11. Le fichier /etc/nagios/objects/commands.cfg contient les définitions des commandes de supervision pouvant être lancées par le service Nagios. Ajouter à la fin de ce fichier la définition de la commande **check_jitter** :

```
define command {
command_name        check_jitter
command_line        /usr/lib/nagios/plugins/check_jitter  $HOSTADDRESS$ $ARG1$ $ARG2$
}
```

- La ligne command_line définit la commande qui sera exécutée par Nagios.
- $ARG1$ et $ARG2$ seront remplacés par la valeur du premier et deuxième argument lorsque nous définirons (point suivant) le service utilisant cette commande.
- $HOSTADDRESS$ sera remplacé par son IP.

12. Modifier le fichier **/etc/nagios/objects/agent.cfg** pour indiquer que nous allons maintenant utiliser la commande définie dans le point précédent sur la machine agent pour surveiller la gigue entre la machine Nagios et la machine agent. Il faut pour cela définir un nouveau **service** comme nous l'avons fait dans la Question 7.
    La commande Nagios que nous avons définie au point précédent prend deux arguments ($ARG1$ et $ARG2$). Pour indiquer les valeurs de ces arguments, il faut les faire précéder par un point d'exclamation, comme ceci :

    ```
    check_command check_jitter !2!10
    ```

    On utilise ici un seuil d'avertissement de 2 ms et un seuil critique de 10 ms.

13. Pour tester, vous utiliserez la commande **tc** (traffic controller) qui permet d'introduire une gigue artificielle sur une interface.
    Par exemple, pour introduire un retard aléatoire de 10±5 millisecondes (en root) :

    ```
    # tc qdisc add dev eth0 root netem delay 10ms 5ms
    ```

    Une fois cette commande exécutée, il faut changer ***add*** par ***change*** si l'on souhaite modifier le retard introduit, ou par ***del*** si l'on souhaite le supprimer.