

## 1) Objectifs :

- Mettre en place et configurer une station de surveillance nagios chargée d'avertir les administrateurs en cas de défaillance sur un des serveurs du réseau.
- Configurer l'utilisation de différents plugin ainsi que certaines extensions de Nagios : nrpe et NSClient qui permettent de surveiller des machines distantes tournant respectivement sous Linux et Windows.

## 2) Remarques :

- Pour chaque machine, ou groupe de machines, il faut préciser les services à surveiller.
- A chaque service correspond un serveur (http,...) ou un plugin Nagios standard.
- Afin d'optimiser la charge réseau, il sera inutile de tester les services d'une machine si celle-ci ne répond pas au ping.

## 3) Installation et configuration de Nagios

- Il est conseillé de réaliser les commandes suivantes avant toute nouvelle installation :

```
sudo apt-get update
sudo apt-get upgrade
```

- Installer **apache2**, **openssh-client**, **openssh-server**, **nagios4**, **nagios-plugins-contrib** et **nagios-nrpe-plugin** de votre distribution Debian.

- Vérifier l'installation avec la commande **sudo /usr/sbin/nagios4stats**

- Pour valider la 'Digest Authentication', commenter les lignes suivantes du fichier **'/etc/apache2/conf-enabled/nagios4-cgi.conf'** :

```
#Require ip .....
#<Files "cmd.cgi">
#Require all granted
#<Files>
```

- Et décommenter :

### **Require valid-user**

- Par défaut Nagios4 ne demande pas de login, ce qui n'est pas conseillé, donc :

- Créer un nouvel utilisateur, par exemple "nagios" :

```
sudo htdigest /etc/nagios4/htdigest.users Nagios4 nagios
```

- Avec la commande suivante, valider les 3 modules d'apache2 suivants:

```
sudo a2enmod authz_groupfile auth_digest cgi
```

- Redémarrer apache2 : **systemctl restart apache2**

- Valider les accès aux CGI pour l'utilisateur précédemment créé (exemple nagios) en modifiant le fichier **'/etc/nagios4/cgi.cfg'**, afin d'y ajouter le nouvel utilisateur:

```
use_authentication=1
authorized_for_system_information=nagiosadmin,nagios
authorized_for_configuration_information=nagiosadmin,nagios
authorized_for_system_commands=nagiosadmin,nagios
authorized_for_all_services=nagiosadmin,nagios
authorized_for_all_hosts=nagiosadmin,nagios
authorized_for_all_service_commands=nagiosadmin,nagios
authorized_for_all_host_commands=nagiosadmin,nagios
```

- Redémarrer apache2 et peut être nagios4 (Test ok)

- Informations diverses :

- Une configuration de base est placée dans le répertoire **/etc/nagios4/objects**.

- Pour configurer Nagios (ajout de machines, contacts et services), il est possible de :

- modifier ou ajouter des fichiers de configuration qui se trouvent dans **/etc/nagios4/objects/**
- modifier le fichier **/etc/nagios4/nagios.cfg** pour activer les nouveaux fichiers que vous créez.

- Certaines commandes sont prêtes et regroupées dans des fichiers **.cfg** du répertoire **/etc/nagios-plugins/config/**

- Il est possible d'ajouter d'autres utilisateurs :

Pour ajouter un nouvel utilisateur : **sudo htdigest /etc/nagios4/htdigest.users philippe philippe**

Pour créer un nouveau fichier : **sudo htdigest -c /etc/nagios4/htdigest.users nagiosadmin**

- Pour tester, utiliser un navigateur web et saisir l'URL : <http://localhost/nagios4>

- Observer le fichier **/etc/nagios4/cgi.cfg**. Choisir au démarrage un mode de vue, par exemple Circular Markup.

## 4) Configurer Nagios pour le réseau LAN :

La configuration par défaut **"/etc/nagios4/nagios.cfg"**.

Dans le fichier **"/etc/nagios4/objects/templates.cfg"** observer la configuration (template : patron) définissant les paramètres par défaut d'un "host".

Vérifier les paramètres utilisés au moment du démarrage et redémarrage de nagios :

```
# Generic host definition template - This is NOT a real host, just a template!
```

```
define host{
    name                generic-host    ; The name of this host template
    notifications_enabled 1            ; Host notifications are enabled
    event_handler_enabled 1            ; Host event handler is enabled
    flap_detection_enabled 1           ; Flap detection is enabled
    process_perf_data     1            ; Process performance data
    retain_status_information 1         ; Retain status information across program restarts
    retain_nonstatus_information 1      ; Retain non-status information across program restarts
    notification_period    24x7
    register              0            ; DON'T REGISTER THIS DEFINITION - ITS NOT A REAL HOST, JUST A TEMPLATE!
}
```

On remarquera la ligne "register 0" qui précise la définition d'un patron.

On observera les autres définitions des patrons avec éventuellement un principe d'héritage(use) : contact, linux-server, window-server, generic-printer, generic-switch, generic-service, local-service

Observation de la configuration de localhost :

- Vérifier que le fichier **"/etc/nagios4/objects/localhost.cfg"** soit activé par le fichier **"/etc/nagios4/nagios.cfg"**
- Dans le fichier "localhost.cfg", observer
  - la définition host avec un héritage sur le patron linux-server
  - l'appartenance au groupe linux-server
  - les services qui lui sont associés : check\_ping, check\_local\_disk, etc.
- Pour tester les modifications de la configuration :  
Recharger les fichiers de configuration Nagios :  
**/etc/init.d/nagios4 reload** ou  
**service nagios4 reload**  
Autoriser les commandes externes :  
**check\_external\_commands=1** dans **/etc/nagios4/nagios.cfg** puis,  
Éventuellement: **sudo usermod -G nagios www-data** et **sudo service apache2 restart**

Vérifiez ensuite, via l'interface web, que le système détecte bien votre configuration. Un temps d'attente est nécessaire.

Ajouter un répertoire pour y ajouter de nouveaux fichiers de configuration

- Toujours dans le répertoire **"/etc/nagios4/"**, créer un sous répertoire **"/routers"**.
- Activer ce répertoire dans la configuration via le fichier **"/etc/nagios4/nagios.cfg"** en lui ajoutant la ligne **"cfg\_dir=/etc/nagios4/routers"**

Définition des différentes Machines correspondant à l'architecture locale (à adapter selon la salle de TP):

- Repérer l'adresse ip et le masque réseau de la station nagios4 (ip a) : exemple 192.168.0.14/24
- Détecter les machines sur le réseau : exemple nmap -v 192.168.0.0/24
- Repérer les stations
  - hôte VirtualBox : exemple 192.168.0.94
  - hôte Gateway : exemple 192.168.0.1
  - etc.
- On créera un fichier par machine.
- Ajouter les hôtes du réseau:
  - Créer le fichier **"/etc/nagios4/routers/gateway.cfg"** et y placer les lignes suivantes :  
# une définition de station pour la passerelle de la route par défaut  
define host {  
    host\_name    **gateway**  
    alias        Default Gateway  
    address      192.168.0.1  
    use          generic-host  
}
  - Ajouter le service de vérification par Ping, dans **"/etc/nagios4/routers/gwservices.cfg"**, ajouter :  
define service {  
    hostgroup\_name    **ping-servers**  
    service\_description  **PING**  
    check\_command      **check\_ping!100.0,20%!500.0,60%**  
    use                generic-service  
    notification\_interval  0  
}
  - Créer des groupes dans le fichier **"/etc/nagios4/objects/hostgroups\_nagios2.cfg"**  
# Un groupe pour les stations pingables  
define hostgroup {  
    hostgroup\_name  ping-servers  
    alias          Pingable servers  
    members        **gateway**  
}  
# Un groupe pour les routeurs  
define hostgroup {  
    hostgroup\_name  **router-servers**  
    alias          Router Servers  
    members        **gateway**  
}  
Si le gateway propose un serveur Web, activer le service test http pour cette machine toujours dans ce fichier  
# Un groupe pour les serveurs web  
define hostgroup {  
    hostgroup\_name  **http-servers**  
    alias          HTTP servers  
    members        **localhost,gateway**  
}
  - Dans le fichier **"/etc/nagios4/objects/services\_nagios2.cfg"**, associons ce groupe à un service :  
# Verifier que les serveurs WEB sont actifs  
define service {  
    hostgroup\_name    **http-servers**  
    service\_description  **HTTP**  
    check\_command      **check\_http**  
    use                generic-service  
    notification\_interval  0  
}
  - Observer le fichier **"/etc/nagios-plugins/config/http.cfg"** :
    - Où se trouve les plugins nagios ?
    - Qu'est-ce qu'un plugin ? : **"file ../ check\_http"**
    - Pour obtenir l'aide d'un plugin par exemple : **"../check\_http --help"**

- Utiliser la commande `check_https` sur un serveur HTTPS en ajoutant un `hostgroup` et un `service`
- Changeons l'image(icon) de ce gateway dans le Status Map de l'interface Web de Nagios4. Dans un nouveau fichier `/etc/nagios4/conf.d/extinfo_nagios.cfg` ajouter les lignes suivantes :

```
define hostextinfo{
    hostgroup_name    router-servers    #ce groupe doit être défini !
    notes             Router servers
    #les images pour  Nagios sont dans /usr/share/nagios/htdocs/images/logos/base/
    icon_image        base/router40.png
    icon_image_alt    Router
    vrm1_image        router40.png
    statusmap_image   base/router40.gd2 jmpI
}
```

- Ajoutons un 'switch1'

- Créer le fichier `/etc/nagios4/objects/switch_nagios.cfg` et y placer les lignes suivantes :

```
define host{
    use                generic-host      ; quel patron?
    host_name          switch1           ; nom du switch
    alias              Switch
    address            xxx.xxx.xxx.xxx   ; adripv4 du switch
}
```

- Ajouter une station 'station1' ou tout autre nom de type Ubuntu connecté à switch1:

- Créer le fichier `/etc/nagios4/objects/station1_nagios.cfg` et y placer les lignes suivantes :

```
define host{
    use                generic-host      ; quel patron?
    host_name          station1         ; nom de la machine
    alias              Ubuntu station
    address            xxx.xxx.xxx.xxx
    parents            switch1
}
```

- Activer le test PING pour cette machine(vu précédemment).
- Activer les tests HTTP et HTTPS pour cette machine si elle propose ces services.

- Créons un groupe pour les stations Ubuntu : dans le fichier `/etc/nagios4/objects/hostgroups_nagios2.cfg`

```
## A list of your Ubuntu GNU/Linux servers
define hostgroup {
    hostgroup_name    ubuntu-servers
    alias             Ubuntu GNU/Linux Servers
    members           station1
}
```

- Créons de nouvelles icônes pour d'autre machine (Ubuntu, WinXP, WinVista):

- Vérifier que les paquetages "libgd-tools" et "netpbm" sont installés.

- Soit le script shell `genicons.sh`

```
#!/bin/bash
path="/usr/bin"
echo "Il est preferable de disposer d'images gif de dimensions 60x60."
echo "Usage: $0 img1.gif [img2 ...]"
for arg
do
    if [ -f "$arg" ]; then
        echo "Conversion de $arg"
        arg=$(echo $arg | sed 's/\.gif$//')
        $path/giftoptnm $arg.gif > $arg.pnm
        $path/pnmtopng -transparent rgb:ff/ff/ff $arg.pnm > $arg.png
        $path/pnmtjpeg -quality=100 -optimize -smooth=0 $arg.pnm > $arg.jpg
        $path/pngtogd2 $arg.png $arg.gd2 0 1
    fi
done
rm -f *.pnm
exit 0
```

- Utiliser ce script sur une image Gif n'excédant pas 60 pixels. On obtient ainsi 4 fichiers à copier dans le répertoire `/usr/share/nagios/htdocs/images/logos/base`

- Sélectionnons son image(icon) dans le Status Map de l'interface Web de Nagios : dans le fichier `/etc/nagios4/objects/extinfo_nagios2.cfg` ajouter les lignes suivantes :

```
define hostextinfo{
    hostgroup_name    ubuntu-servers
    notes             Ubuntu GNU/Linux servers
    #/usr/share/nagios3/htdocs/images/logos/base/.....png
    icon_image        base/ubuntu.png
    icon_image_alt    Ubuntu GNU/Linux
    vrm1_image        ubuntu.png
    statusmap_image   base/ubuntu.gd2
}
```

- Ajouter des machines Windows XP

- Ajouter un groupe "xp-servers" dans le fichier `/etc/nagios4/objects/hostgroups_nagios.cfg`

- Ajouter aussi ce groupe "xp-servers" dans le fichier `/etc/nagios4/objects/extinfo_nagios.cfg`. Créer les fichiers images pour ce type de station.

- Créer le fichier `"/etc/nagios4/objects/stationxp1_nagios.cfg"` et y placer les lignes suivantes :

```
define host{
    use                generic-host        ; quel patron?
    host_name          stationxp1          ; nom de la machine
    alias              Windows XP Station
    address            xxx.xxx.xxx.xxx
    parents            xxxxxx
}
```

- Ajouter la station dans le service ping, http si nécessaire...

## 5) Configurer Nagios pour le réseau WAN :

Ajoutons une passerelle "gatewayiris" après le "gateway" :

- Créer le fichier `"/etc/nagios4/objects/gatewayiris_nagios2.cfg"`
- Y déclarer un host "gatewayiris" d'adresse "XXX.XXX.XXX.XXX" en précisant qu'il est placé après "gateway" en ajoutant la ligne "parents gateway"
- Tester la commande ping sur ce routeur et dans le cas d'une réponse positive, l'ajouter dans le service ping de Nagios.

Ajoutons des stations après la passerelle "gatewayiris" :

- Créer le fichier `"/etc/nagios3/conf.d/kwartzirisdk_nagios2.cfg"`
- Y déclarer un host "kwartzirisdk" d'adresse "XXX.XXX.XXX.XXX" en précisant qu'il est placé après "gatewayiris"
- Pour accéder à ce serveur HTTP, on utilisera la port 8080 dont le service Nagios n'existe pas encore.
- Ajoutons le service "http:8080" grâce au paragraphe suivant

## 6) Activer de nouveaux services

Configurons le monitoring de services ne nécessitant pas d'autres plugins que ceux installés par défaut avec Nagios.

- Ajoutons le service de test "http:8080"
  - tester une des commandes suivantes :
 

```
/usr/lib/nagios/plugins/check_http -p proxyPort -I proxyIPAdr -b userProxy:pwd -u http://serveurHTT:8080
```

```
/usr/lib/nagios/plugins/check_http -p 8080 -I serveurHTTP
```

 Voir `--help` pour l'aide en ligne. Le résultat devrait être :
 

```
"HTTP OK : HTTP/1.1 200 OK - ..."
```
  - Et donc, dans le fichier `"/etc/nagios3/conf.d/service_nagios2.cfg"` ajouter les lignes suivantes :
 

```
define service {
    hostgroup_name          http-servers8080
    service_description     HTTP8080
    check_command           check_http!-p 8080 -I .....
    use                     generic-service
    notification_interval   0 ; set > 0 if you want to be renotified
}
```
  - Créons un nouveau groupe pour cette station : dans le fichier `"/etc/nagios3/conf.d/hostgroups_nagios2.cfg"`

```
define hostgroup {
    hostgroup_name    http-servers8080
    alias             http8080
    members           kwartzirisdk
}
```
  - Tester ce nouveau service
- Ajoutons le service de test "dhcp" connaissant votre serveur dhcp ou après avoir installer un serveur dhcp sur localhost.
  - tester la commande :
 

```
sudo /usr/lib/nagios/plugins/check_dhcp -s leServeurDHCP -i eth0
```

 Le résultat pourrait être :
 

```
"Erreur: Impossible de connecter le socket à l'interface eth0. Vérifiez vos droits..."
```

 ou
 

```
"OK: Reçu 1 DHCPOFFER(s), bail maximum = 43200 sec." avec "sudo"
```
  - Observer le fichier `"/etc/nagios-plugins/config/dhpc.cfg"`
  - Observer Dans le fichier `"/etc/nagios3/conf.d/service_nagios2.cfg"` ajouter les lignes suivantes :

```
define service {
    hostgroup_name          dhcp-servers-eth0
    service_description     DHCP eth0
    check_command           check_dhcp-interface!eth0
    use                     generic-service
    notification_interval   0 ;
}
```

- Créons un nouveau groupe pour les serveur DHCP : dans le fichier `"/etc/nagios3/conf.d/hostgroups_nagios2.cfg"`

```
define hostgroup {
    hostgroup_name    dhcp-servers-eth0
    alias             DHCP Servers sur eth0
    members           xxxxxxxx,yyyyyy
}
```
- Tester ce nouveau service.
- Il est peut-être nécessaire de donner plus de droit à Nagios : `chmod u+s /usr/lib/nagios/plugins/check_dhcp`
- Pourquoi ne pas faire la même chose sur wlan0 correspondant au Wifi

Tester le plugin SNMP :

```
/usr/lib/nagios/plugins/check_snmp -H localhost -C public -o 1.3.6.1.2.1.1.0
                                     ou -o system.sysDescr.0
```

```
/usr/lib/nagios/plugins/check_snmp -H gateway -C public -o interfaces.ifTable.ifEntry.ifDescr.1
```

ou -o 1.3.6.1.2.1.2.2.1.2.1

Exemple de réponse

SNMP OK - loopback | IF-MIB::ifDescr.1=loopback

- En complétant les fichiers services.cfg et commands.cfg afin de tester les services disponibles sur une station linux, la station Nagios, un routeur adsl(gateway), ainsi que tous les services PING.
- Dans le fichier commands.cfg, définir les commandes à exécuter pour tester un service. L'ensemble de ces commandes est présent sous forme d'exécutable dans le répertoire plugin de Nagios. On peut donc tester en ligne de commande si les paramètres associés aux commandes fonctionnent. De même, on obtient l'aide relative à cette commande grâce à l'option -h (ex: check\_ping -h).

Exemple de service à tester : pour le test du SNMPGet, voici la définition à ajouter dans /etc/nagios3/commands.cfg:

```
define command{
    command_name    check_snmp
    command_line    $USER1$/check_snmp -H $HOSTADDRESS$ -C $ARG1$ -o $ARG2$$ARG3$
}
```

Ainsi, pour associer le test du service snmp, on utilisera le nom check\_snmp. Deux solutions :

- soit on indique "en dur" les paramètres de la fonction check\_snmp : BOF
- soit on indique les paramètres à utiliser et ces derniers seront écrits dans le fichier services.cfg

Certains sont rentrés en dur, comme l'adresse de l'hôte que l'on récupèrera dans le hostgroup. D'autres seront à entrer en complément de la commande, comme ARG1 pour la communauté, ARG2 pour le noeud OID et ARG3 éventuellement pour l'instance de l'OID.

Il faut configurer des commandes non définies dans commands.cfg mais présentes dans le répertoire de plugin. Ainsi, on rajoute la définition de check\_ssh, check\_daytime et check\_swap (voir fichier commands.cfg).

- Dans le fichier services.cfg, on associe aux hôtes déjà définis le service que l'on souhaite tester ainsi que les arguments nécessaires à son utilisation. Exemple du SNMPGet d'une station :

```
define service{
    use                generic-service
    hostgroup_name     agents-snm
    service_description SNMP system.sysDescr
    notification_interval 240
    check_command      check_snmp!public!1.3.6.1.2.1.1.0
    #OU check_command  check_snmp!public!sysDescr.0 #le noeud et l'instance regroupes dans $ARG2$
    #OU check_command  check_snmp!public!sysDescr!.0 #le noeud et l'instance dissociés $ARG2$$ARG3$
}
```

- Créons le nouveau groupe pour les agents SNMP : dans le fichier "/etc/nagios3/conf.d/hostgroups\_nagios2.cfg"

```
define hostgroup {
    hostgroup_name agents-snm
    alias Agents SNMP
    members gateway
}
```

On configure ici l'hôte sur lequel on teste le service, les périodes de test et les périodes de notification ainsi et surtout la commande à exécuter avec ses paramètres (ceux non rentrés en durs).

Les arguments de la commande sont séparés par des "!", dans le cas du SNMPget, l'argument 1 correspondant à ARG1 dans commands.cfg et vaut 'public', de même l'argument 2 (ARG2) vaut '1.3.6.1.2.1.1.0'. Ces paramètres dépendent de la commande exécutée.

Supposons que gateway a une ligne ADLS sur l'interface indexée 4. Nous désirons afficher son état Up/Down/Testing/Unknow...

- Ajoutons le service qui permet de connaître l'état d'une interface :

```
define service{
    use                generic-service
    hostgroup_name     agents-snm-adsl
    service_description SNMP adsl ifOperStatus
    notification_interval 240
    #check_command     check_snmp!public!1.3.6.1.2.1.2.2.1.8.4
    #check_command     check_snmp!public!ifOperStatus.4
    check_command      check_snmp!public!ifOperStatus!.4
}
```

Ajoutons le groupe associé à ce service :

```
define hostgroup {
    hostgroup_name agents-snm-adsl
    alias Agents SNMP ifOperStatus
    members gateway
}
```

## 7) Utilisation de NSClient

Surveiller les services disponibles sur une station windows. Pour cela :

- utiliser le serveur NSClient à installer sur la machine windows,
  - ainsi que le plugin check\_nt, disponible avec NSClient, afin d'interroger le serveur.
  - compléter les fichiers services.cfg et checkcommands.cfg en conséquences.
- Une fois l'archive zip décompressée, il faut copier les fichiers pNSClient.exe, pdh.dll, psapi.dll et counters.defs dans le répertoire où l'on veut installer NSClient (par exemple c:\nsclient) puis lancer la commande pNSClient.exe /install dans un prompt MS-DOS, ce qui installe le service NSClient sur la machine.  
On peut alors démarrer le service.
  - Du côté de la machine Nagios, il faut utiliser le plugin check\_nt qui est fourni dans le fichier zip et le copier dans le répertoire des plugins de Nagios.

- On modifie le fichier checkcommands.cfg en y ajoutant les blocs suivant:

- Test de la mémoire utilisée:

```
define command{
    command_name    check_windows_mem
    command_line    $USER1$/check_nt -H $HOSTADDRESS$ -v MEMUSE -w $ARG1$ -c $ARG2$
}
```

- Test de l'espace disque utilise :

```
define command{
    command_name    check_windows_disk
    command_line    $USER1$/check_nt -H $HOSTADDRESS$ -v USEDDISKSPACE -l $ARG1$ -w $ARG2$ -c $ARG3$
}
```

- Puis on ajoute les blocs suivant dans le fichier services.cfg pour utiliser ces commandes sur la machine Windows :

- Test de la mémoire utilisée :

```
define service{
    use                generic-service           ; Name of service template to use
    host_name          Non_de_la_Station
    service_description MEMORY_USED
    is_volatile        0
    check_period       24x7
    max_check_attempts 3
    normal_check_interval 5
    retry_check_interval 1
    contact_groups     Windows
    notification_interval 240
    notification_period 24x7
    notification_options c,r
    check_command      check_windows_mem!50%!90%
}
```

- Test de l'espace disque disponible:

```
define service{
    use                generic-service           ; Name of service template to use
    host_name          Non_de_la_Station
    service_description DISK
    is_volatile        0
    check_period       24x7
    max_check_attempts 3
    normal_check_interval 5
    retry_check_interval 1
    contact_groups     Windows
    notification_interval 240
    notification_period 24x7
    notification_options c,r
    check_command      check_windows_disk!c!60%!80%
}
```