TP7 : Utilisation de SNMP

Nagios®

Table des matières

Objectif : Ce document va expliquer comment utiliser le protocole SNMP pour superviser des équipements.


INFO : C'est juste de l'information mais qui peut être nécessaire à la compréhension du TP


IMPORTANT : Information importante à prendre en compte


TP à Faire : Travaux à réaliser

TP7 : Utilisation de SNMP

Nous allons nous appuyer sur le protocole SNMP pour contrôler certains services et pour envoyer des traps.

Rappel : Simple Network Management Protocol (abrégé **SNMP**), est un protocole de communication qui permet aux administrateurs réseau de gérer les équipements du réseau, de superviser et de diagnostiquer des problèmes réseaux et matériels à distance.

SNMP : Contrôle Actif

7.1 - Configuration d'un serveur distant sous Linux pour activer SNMP


- Installer le service *snmpd* :

```
# apt-get install snmp snmpd
```


- Ajouter les lignes suivantes dans le fichier */etc/snmp/snmpd.conf* :

Fichier snmpd.conf

```
# Créer le "security name" nagios associé à la communauté SNMP communauteNagios
pour le sous-réseau 172.30.64.0/20
com2sec nagios 172.30.64.0/20 communauteNagios

# Associer au "security name" nagios le groupe collecte
group collecte v1 nagios

# La vue tout permet de parcourir l'ensemble de la MIB
view tout included .1

# Le groupe collecte a accès en lecture à la vue tout
access collecte "" any noauth exact tout none none
```

 - Modifier le fichier /etc/default/snmpd et remplacer la ligne suivante :

Fichier snmpd

```
SNMPDOPTS='-Lsd -Lf /dev/null -u snmp -g snmp -I -smux -p /var/run/snmpd.pid
127.0.0.1'

Remplacer par

SNMPDOPTS='-Lsd -Lf /dev/null -u snmp -g snmp -I -smux -p /var/run/snmpd.pid'
```

Désormais snmpd est en écoute sur toutes les interfaces.

 - (Re)lancer le service snmpd :

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

 - Vérifier depuis la machine Nagios l'accès aux informations :

```
# snmpwalk -v 1 -c communauteNagios machine_cible .1
```

 *La MIB standard devrait apparaître…Si vous avez une erreur du genre : **Timeout: No Response from x.x.x.x.** vous devez modifier un paramètre dans le fichier de configuration "**/etc/snmp/snmpd.conf**"…mais lequel?*

 **Compte rendu-13 : Quelle modification avez-vous apportez dans le fichier snmpd.conf pour que la commande snmpwalk ci-dessus fonctionne et pourquoi?**

7.2 – Configuration d'un routeur Cisco pour activer SNMP

 - Définir les champs contact et location de l'agent SNMP du routeur :

```
# enable
# conf t
router(config)#snmp-server location « Routeur DMZ »
router(config)#snmp-server contact admin@domain.fr
```

 - Configurer le routeur afin qu'il accepte les get SNMP avec la communauté communauteNagios :

```
router(config)#snmp-server community communauteNagios RO
```

7.3 - Utilisation des plugins SNMP pour interroger un serveur ou routeur distant

 - Utiliser le check_snmp en ligne (exemple) :

```
# ./check_snmp -P 1 -H serveur-linux -o .1.3.6.1.2.1.2.2.1.8.1 -C communauteNagios
-s 1 -l "interface 1"
```

Permet de voir si l'interface 1 est up

 - Définir cette commande dans commands.cfg :

Fichier commands.cfg

```
define command      {
       command_name         check_snmp_interface_updown
       command_line         $USER1$/check_snmp -P 1 -H $HOSTADDRESS$ -o
.1.3.6.1.2.1.2.2.1.8.$ARG1$ -C communauteNagios -r '1' -l "Interface $ARG2$ (SNMP)
       }
```

-P : version de snmp
-o : OID
-C : Communauté
-r : REGEX, retourne OK si la valeur est 1 sinon Critical
-l : description

Le travail peut devenir très fastidieux de trouver l'Oid correspondant à l'information recherchée. Par chance, quelqu'un a déjà fait le travail pour nous en développant une série de plugins SNMP que nous allons utiliser!

 - Installer les plugins SNMP de Manubulon :

```
# sudo apt-get install nmap librrds-perl libgd-gd2-perl libnet-snmp-perl
# cd /usr/src
# wget http://nagios.manubulon.com/nagios-snmp-plugins.1.1.tgz
# tar xzf nagios-snmp-plugins.1.1.tgz
# cd nagios_plugins
# ./install.sh

Le script se termine par :

Installation                                completed                                OK
You      can      delete      all      the      source      files      and      directory
Remember to look for informtation at http://www.manubulon.com/nagios
```

Compte rendu-14 : Créer les commandes et les services pour contrôler chacune des partitions, la charge CPU …de serveur-linux en utilisant les plugins perl check_snmp* de Manubulon.
Nommer vos services *-snmp pour les différencier des autres services.

SNMP : Contrôle Passif

7.4 - Configuration d'un serveur distant sous Linux pour envoyer des traps SNMP

- Editer le fichier snmpd.conf et ajouter les lignes suivantes à la fin du fichier :

Fichier snmpd.conf

```
# Définit la communauté par défaut à utiliser quand des traps seront envoyées.
trapcommunity communauteNagios

# Envoi de traps de version 1 - HOTE où se trouve le daemon snmptrad [Community [Port]]
trapsink IP-serveurNagios communauteNagios 162

# Envoi des traps de version 2
trap2sink IP-serveurNagios communauteNagios 162

#Utilisé lors d'envois d'information â la place des traps
informsink IP-serveurNagios communauteNagios 162

# Envois de traps lors d'authentification ratée
authtrapenable 1

# Permet d'envoyer des traps lors du changement d'état (Down/Up) des interfaces réseaux
linkUpDownNotifications yes

# Envois des traps pour une liste de problèmes généraux
defaultMonitors yes
```

*Remplacer **IP-serveurNagios** par l'adresse IP de votre serveur Nagios.*

 - Tester si l'envoi de traps vers le serveur Nagios se fait bien :

 - Lancer un tcpdump sur la machine Nagios pour voir si les Traps arrivent bien :

```
# tcpdump –i eth0 port 162
```

 - Relancer le service snmpd (ca doit générer des traps de type coldStart) :

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

Vous devriez voir une trap de type coldStart passer (voir la signification des traps à la fin du Tp) :

```
# tcpdump –i eth0 port 162
13:36:40.285251 IP debian-client.39490 > debian-server.local.snmp-trap:  C=communauteNagios
Trap(29)  E:8072.3.2.10 10.0.2.15 coldStart 9
```

7.5 - Configuration d'un routeur Cisco pour envoyer des traps SNMP

 - Définir l'adresse de destination pour les traps de la communauté communauteNagios :

```
router(config)#snmp-server host serveurnagios communauteNagios snmp
```

 - Définir l'interface Ethernet comme interface source des alarmes :

```
router(config)#snmp-server trap-source ethernet0
```

 - Activer les alarmes SNMP pour authentication linkdown linkup coldstart :

```
router(config)#snmp-server enable traps snmp authentication linkdown linkup coldstart
warmstart
```

7.6 – Configuration de Nagios pour analyser les traps envoyées par un serveur Linux ou un routeur Cisco.

Dans notre TP nous allons simuler la défaillance d'une interface réseau sur un serveur Linux et être alerté de ce problème via Nagios.

 Fonctionnement général :

- Un hôte sur le réseau envoie une interruption SNMP au serveur Nagios (exemple : arrêt d'une interface réseau). Celui-ci la reçoit via le service snmptrapd (sur le port UDP 162).
- Snmptrapd la passe ensuite à SNMPTT, dont le rôle est de rendre intelligible l'interruption.
- SNMPTT envoie l'interruption interprétée à Nagios, via le fichier de commandes externes de celui-ci. Il utilise la commande : submit_check_result

7.6.1 – Installation de SNMPTT

 - Compléter la couche SNMP et librairie perl pour faire fonctionner SNMPTT.

```
# sudo apt-get install snmpd libconfig-inifiles-perl
# wget http://switch.dl.sourceforge.net/sourceforge/snmptt/snmptt_1.4.tgz
# tar -xzf snmptt_1.4.tgz
# cd snmptt_1.4/
# sudo mv snmpttconvert snmpttconvertmib snmptt-net-snmp-test /usr/local/bin/
# sudo mv snmptt snmptthandler /usr/local/sbin/
# sudo mkdir -p /usr/local/etc/snmptt.d
# sudo cp snmptt.ini /usr/local/etc/
# sudo cp examples/snmptt.conf.generic /usr/local/etc/snmptt.d/
```

7.6.2 – Activation et configuration de snmptrapd

 - Modifier le fichier /etc/default/snmpd pour activer le démon snmptrapd :

Fichier snmpd

```
TRAPDRUN=yes

# snmptrapd options (use syslog).
TRAPDOPTS='-Lsd -On -p /var/run/snmptrapd.pid'
```

 Le lanceur du service snmptrapd doit être modifié pour ne pas traduire les OID, mais les laisser sous forme numérique. C'est SNMPTT qui fera la traduction.

 - Modifier le fichier de configuration /etc/snmp/snmptrapd.conf :

Fichier snmptrapd.conf

```
traphandle default /usr/local/sbin/snmptt
disableAuthorization yes
donotlogtraps  yes
```

 Explications :

- on traite de la même façon toutes les interruptions : avec SNMPTT
- on accepte toutes les interruptions
- on ne journalise pas les interruptions reçues (c'est SNMPTT qui s'en chargera). On aura quand même une trace de snmptrapd dans syslog, grâce à l'option -Lsd du lanceur.

7.6.3 – Compilation des Mibs

 Il faut maintenant récupérer le ou les fichiers MIB des équipements à superviser, pour les convertir au format SNMPTT. Le but est d'extraire ce qui dans le fichier MIB est un commentaire, pour que SNMPTT le mette en correspondance avec l'OID qui sera reçu. Au préalable il nous faut ajouter un nouveau dépôt dans le fichier sources.list.

 - Ajouter deux nouveaux dépôts dans le fichier /etc/apt/sources.list

Fichier sources.list

```
deb http://ftp.fr.debian.org/debian/ wheezy contrib non-free
deb-src http://ftp.fr.debian.org/debian/ wheezy contrib non-free
```

 - Mettre à jour la liste des fichiers disponibles à partir des nouveaux dépôts puis installer les fichiers Mibs.

```
# apt-get update

Puis installer le paquet snmp-mibs-downloader

# apt-get install snmp-mibs-downloader
```

 - Récupérer également la commande **`submit_check_result`** dans les sources de nagios (voir TP1) et la copier dans le répertoire des plugins :

```
# cp /usr/local/src/nagios/contrib/eventhandlers/submit_check_result
/usr/local/nagios/libexec/
```

 - Pour chaque fichier MIB, on lance la commande :

```
# snmpttconvertmib --in=<fichier MIB> --out=/etc/snmp/snmptt.conf.<equipement> \
--exec='/usr/local/nagios/libexec/submit_check_result $R TRAP 1'
```

<fichier MIB> = /var/lib/mibs/ietf/IF-MIB (c'est le fichier MIB qui nous intéresse pour cet exercice)
<equipement> = nom à donner pour différencier les fichiers snmptt.conf

- On obtiendra dans le fichier résultat des blocs de configuration pour SNMPTT, un par OID, qui ressemblent à ça :

Fichier snmptt.conf.<equipement>

```
FORMAT $*
EXEC /usr/local/nagios/libexec/submit_check_result $R TRAP 1 "$*"
SDESC
La description.
EDESC
```

 - Explication de la commande **submit_check_result $r TRAP 1 "$*" :**

```
/usr/local/nagios/libexec/submit_check_result;$1;$2;$3;$4

# Arguments:
# $1 = host_name (nom de l'hôte auquel le service est associé)
# $2 = svc_description (Description du service)
# $3 = return_code (Un entier qui détermine l'état du service, 0=OK, 1=WARNING, 2=CRITICAL,
3=UNKNOWN).
# $4 = plugin_output (Texte)
```

 Dans le fichier généré, vous allez faire quelques modifications, remplacer la description du service TRAP par LINK-eth0 qui sera la description de notre service Nagios. Et remplacer la valeur de l'argument $3 par 2 (CRITICAL) pour la partie linkDown et 0 (OK) pour la partie linkUp.

 - Ajouter enfin au fichier de configuration /usr/local/etc/snmptt.ini les fichiers générés :

Fichier snmptt.ini

```
[...]
[TrapFiles]
snmptt_conf_files = <<END
/etc/snmp/snmptt.conf.<equipement>
```

```
/etc/snmp/snmptt.conf.<equipement1>
/etc/snmp/snmptt.conf.<equipement2>
END
```

7.6.4 - Définition d'un service passif

Le principe est d'utiliser, pour recevoir les interruptions SNMP, des services passifs mais aussi volatils. C'est à dire que le démon nagios attend les informations que les clients lui transmettent. Ce n'est pas lui, de sa propre initiative qui va demander une information (contrairement à un plugin normal ou au plugin NRPE).

 - Définir un template pour les traps SNMP :

Fichier templates.cfg

```
define service{
   name                    snmptrap-service
   use                     generic-service
   service_description     LINK-eth0
   is_volatile             1
   check_command           check-host-alive
   max_check_attempts      1
   normal_check_interval   1
   retry_check_interval    1
   passive_checks_enabled  1
   check_period            24x7
   notification_interval   31536000
   contact_groups          groupe-systeme-reseau
   register                0
}
```

 **Compte rendu-15 : A quoi sert la commande check-host-alive ici ?**

 - Attribuer ce template à chaque machine supervisée :

Fichier serveurs-linux.cfg

```
define service{
   host_name               serveur-linux
   use                     snmptrap-service
   contact_groups          groupe-systeme-reseau
}
```

 - Recharger Nagios pour prendre en compte les modifications :

```
# /etc/init.d/nagios reload
```

7.6.5 - Test

 - Tester si l'envoi de traps vers le serveur Nagios se fait bien : lancer un tcpdump sur la machine Nagios pour voir si les traps arrivent bien :

```
# tcpdump –i eth0 port 162
```

 - Tester la réception et l'interprétation de traps SNMP par Nagios : désactiver une interface réseau pour l'envoi de traps (ça doit générer des traps de type linkDown) :

```
# ifconfig eth0 down
```

La désactivation d'une interface réseau génère une trap de type linkDown

- Vous devez voir la trap **linkDown** passer, qui signifie que l'interface réseau est tombée.

```
# tcpdump –i eth0 port 162
01:48:09.145935 IP debian-client.local.46949 > debian-server.local.snmp-trap:
C=communauteNagios Trap(108)  E:8072.3.2.10 192.168.56.102 linkDown 150025
```

- Si tout a bien été configuré, le service LINK-eth0 doit passer CRITICAL et une notification sera envoyée aux contacts de ce service.

Service	Status	Last Check	Duration	Attempt	Status Information
LINK-eth0	CRITICAL	12-01-2013 13:04:39	0d 0h 0m 7s	1/1	A linkDown trap signifies that the SNMP entity, acting in 2 2 2

 - Faire également un multitail sur les logs de nagios et de snmptt pour suivre les traces :

```
# multitail –f /usr/local/nagios/var/nagios.log –f /var/log/snmptt.log
```

 Pour que ça marche vous devez définir l'adresse IP des clients dans le fichier /etc/hosts du serveur Nagios.

 Pour rétablir la situation, réactiver l'interface réseau :

```
# ifconfig eth0 up
```

 **Compte rendu-16 : Maintenant que vous savez mettre en place un système de Traps sous Linux,**

Serveur Linux :

Vous allez mettre en place un système d'alerte lorsque qu'on vient interroger le client snmp d'un serveur Linux avec une mauvaise communauté :

Exemple : à partir du serveur Nagios faire un ./check_snmp –H serveur-linux –C mauvaise communauté…(communauté différente de celle définie dans le snmpd.conf du serveur-linux)
Le serveur serveur-linux va générer une Trap de type Authentication Failure à cause de la mauvaise communauté.
Vous devez compiler la MIB correspondante et créer un service sous Nagios pour en être alerté.

Aide : la MIB dont vous aurez besoin est SNMPv2-MIB

Si vous avez du temps :

Routeur Cisco :

- Compiler la MIB correspondante pour accepter les traps linkUp et linkDown du routeur Cisco : IF-MIB (déjà fait précédemment).
- Créer le service associé.
- Tester en sélectionnant l'interface et faire un shut et no shut.

La requête d'état arrêt va mettre le service critique, il faut attendre une requête de retour pour qu'il repasse en OK.

 Rappel sur les traps SNMP :

Le trap **Cold Start** signifie que l'agent SNMP a été démarré.

Le trap **Warm Start** signifie que l'agent SNMP a été redémarré suite à une reconfiguration.

Le trap **Link Down** signifie qu'un lien (interface réseau) du host est passé en anomalie ou a été inactivé.

Le trap **Link Up** signifie qu'un lien (interface réseau) du host a été activé ou est de retour en état de fonctionnement.

Le trap **Authentication Failure** signifie qu'une tentative d'accès par SNMP à l'agent SNMP du host a été réalisée avec une communauté SNMP invalide.