TP 2 — Surveillance de la MIB et notifications SNMP

Ce TP est à faire sous marionnet.: une machine agent (qui est supervisée) ; et un NMS (qui supervise). En fin de séance, vous enverrez en guise de compte-rendu :
— Les fichiers de configuration de l'agent (/etc/snmp/snmpd.conf) et du NMS (/etc/snmp/snmptrapd.conf) ; — le script (/bin/traitement-notification) ;
— et un fichier PDF contenant les copies d'écran des notifications envoyés par l'agent.

Exercice 1 — Travail préparatif

I 1.1 *Debian* sur les deux postes.
I 1.2 Sur les deux postes : installez les paquets `snmp`, `snmp-mibs-downloader` et `wireshark`.
I 1.3 Sur l'agent : installez le paquet `snmpd` (qui installe le service homonyme).
I 1.4 Sur le NMS : installez le paquet `snmptrapd` (qui installe le service homonyme).
I 1.5 Sur les deux postes : commentez la ligne `mibs :` dans le fichier `/etc/snmp/snmp.conf`.

La dernière opération permet que les commandes Net-SNMP utilisent les MIBs téléchargées et affichent les noms des objets plutôt que leurs OIDs (p.ex., sysName au lieu de 1.3.6.1.2.1.1.5).

Le fichier de configuration de `snmpd` (l'agent qui répond aux requêtes du NMS) est `/etc/snmp/snmpd.conf`.

I 1.6 Sur la machine agent, supprimez puis recréez ce fichier avec le contenu minimal ci-dessous :

```
agentAddress udp:161
rocommunity public
rwcommunity private
```

Exercice 2 — Surveillance des processus

Il est possible avec les outils Net-SNMP de déclencher l'envoi de notifications suite à la modification des objets de la MIB. Pour pouvoir observer les données relatives aux processus, aux disques durs, à la charge du système, …, les outils Net-SNMP utilisent la MIB UCD (nommée ainsi car initialement conçue à l'Université de Californie, Davis). Cette une MIB privée dont l'OID est 1.3.6.1.4.1.2021.

Nous allons voir dans cet exercice comment, grâce à cette MIB, observer un service en exécution et envoyer une notification en cas d'arrêt du service. Nous prendrons le service `sshd` comme exemple.

I 2.1 Définissez, dans le fichier de configuration de l'agent, une communauté pour l'envoi de notifications ainsi que la destination de ces notifications, comme nous l'avions fait dans le TP précédent :

```
trapcommunity <communaute-de-notification>
trap2sink <IP-du-NMS>
```

Pour la communauté, vous choisirez un nom quelconque.

I 2.2 Ajoutez les quatre lignes ci-dessous dans le fichier de configuration de l'agent :

```
createUser      user
iquerySecName   user
agentSecName    user
rouser          user
```

(Ces lignes créent un utilisateur SNMPv3 appelé `user` ayant le droit d'interroger l'agent. Ces définitions sont nécessaires même si dans ce TP nous utilisons uniquement SNMPv2c.)

Le fichier `/lib/systemd/system/snmpd.service` de l'agent détermine la façon dont est lancé et arrêté le service `snmpd`. Nous allons le modifier car, dans l'état actuel, il ne permet pas d'activer la surveillance de la MIB.

I 2.3 Remplacez dans ce fichier la ligne `Environment="MIBS="` par `Environment="MIBS=ALL"`. (Ignorez ce point si la ligne n'est pas présente.)
I 2.4 Dans ce même fichier, enlevez, dans la ligne `ExecStart=...`, le paramètre `-I -smux,mteTrigger,mteTriggerConf`.
I 2.5 Exécutez la commande ci-dessous pour que les modifications soient prises en compte :

```
# systemctl daemon-reload
```

On peut maintenant configurer la surveillance du processus `sshd`.

I 2.6 Ajoutez la ligne ci-dessous dans le fichier de configuration de l'agent :

```
proc sshd
```

Cette ligne signifie qu'il faut qu'un processus nommé `sshd` soit toujours présent sur la machine.

I 2.7 Démarrez le service `sshd` s'il n'est pas lancé.

I 2.8 Récupérez, avec `snmpwalk`, la branche 1.3.6.1.4.1.2021.2. N'utilisez pas l'option `-On` qui affiche les OIDs sous forme numérique afin de voir les noms des objets récupérés. La branche récupérée devrait contenir des objets créés suite à l'utilisation de la directive `proc`. On voit normalement le nom du processus observé (`prNames`) ainsi qu'un *drapeau* d'erreur (`prErrorFlag`) qui vaut 0 (⇔ pas d'erreur) puisque `sshd` est en exécution.

La supervision d'un service se fait par l'ajout d'un *moniteur*.

I 2.9 Ajoutez les lignes ci-dessous dans le fichier de configuration de l'agent :

```
notificationEvent trapService 1.2.3.1.4.1.1000.10.1 -o prNames -o prErrMessage
monitor -r 10 -e trapService "erreur service" prErrorFlag != 0
```

La seconde ligne peut s'interpréter de cette façon : toutes les 10 secondes (`-r 10`), tester si l'objet `prErrorFlag` est différent de 0 et, si c'est le cas, déclencher l'événement `trapService`. La chaîne de caractère `"erreur service"` est un nom associé au moniteur. L'événement `trapService` est défini à la première ligne. Il consiste à envoyer une notification SNMP ayant l'OID 1.2.3.1.4.1.1000.10.1 et contenant le nom du processus arrêté (`-o prNames`) et le message d'erreur (`-o prErrMessage`), c'est-à-dire les objets renvoyés par la commande `snmpwalk` utilisée précédemment. (*Remarque.* La notification 1.2.3.1.4.1.1000.10.1 n'existe pas dans la MIB. Il faudrait en théorie la déclarer dans la MIB avec le langage de description SMI vu en cours.)

I 2.10 Tout en capturant les trames, arrêtez le serveur `sshd` sur l'agent. On devrait normalement capturer la notification.

La directive `proc` permet aussi de préciser le nombre de processus identiques (ayant le même nom) que l'on souhaite avoir en exécution. Comme le nombre de processus `sshd` dépend du nombre de connexions SSH ouvertes sur la machine (c'est exactement $1 + 2\times$ le nombre de connexions) on peut utiliser ce mécanisme pour recevoir une notification s'il y a trop de connexions ouvertes simultanément. Pour pouvoir tester simplement, on configurera `snmpd` de sorte qu'une notification soit envoyée à la première connexion ouverte (donc avec au maximum 1 processus `sshd`).

I 2.11 Redémarrez le service `sshd`.

I 2.12 En consultant le manuel de `snmpd.conf`, trouvez comment préciser que l'on souhaite toujours avoir un et un seul processus `sshd`.

I 2.13 Vérifier qu'une notification est bien envoyée lorsque l'on ouvre connexion SSH depuis le NMS sur la machine agent (`ssh <ip-de-agent>`). En effet, à ce moment là, le nombre de processus `sshd` en exécution sur l'agent sera de 3, ce qui devrait provoquer l'envoi de la notification.

Exercice 3 — Surveillance du disque

Il est aussi possible de surveiller la taille des fichiers ou des répertoires. Cela peut être utile, par exemple, pour s'assurer que le fichier d'une base de données n'occupe pas trop de place sur le disque. On utilise alors la directive `file`.

I 3.1 Ajoutez dans le fichier de configuration de l'agent la ligne ci-dessous :

```
file <chemin-absolu-du-fichier> <taille-max-du-fichier-en-kilo-octets>
```

Choisissez le chemin d'un fichier inexistant (nous le créerons par la suite) et une taille pas trop grande.

Comme précédemment, des objets donnant des informations sur le fichier observé ont été créés dans la MIB.

I 3.2 Vérifiez, avec `snmpwalk` que la branche 1.3.6.1.4.1.2021.15 est bien présente avec les informations adéquates.

I 3.3 En vous inspirant de ce qui a été fait au point I 2.9 de l'exercice précédent et en observant le résultat de la commande `snmpwalk` utilisé au point précédent, faites en sorte qu'une notification soit envoyée quand la taille du fichier excède la taille maximale fixée. La notification devra contenir trois objets : le nom du fichier observé, sa taille actuelle et le message d'erreur. Observer le résultat de la commande `snmpwalk` pour trouver le nom de ces objets. Comme dans l'exercice précédent, utilisez un OID de notification inexistant.

I 3.4 Testez et vérifiez avec wireshark que la notification a bien été envoyée.

Exercice 4 — Traitement des notifications sur le NMS

Sous Linux, il existe le service `snmptrapd` pour traiter les notifications reçues. Son fichier de configuration est `/etc/snmp/snmptrapd.conf`. Nous allons le configurer pour pouvoir traiter les notifications reçues par le NMS.

I 4.1 Sur le NMS, modifiez le fichier de configuration de `snmptrapd` pour qu'il contienne uniquement la ligne ci-dessous :

```
authCommunity execute <communaute-de-notifications>
traphandle default /bin/traitement-notification
```

La première ligne signifie qu'à chaque notification reçue provenant de la communauté précisée, le NMS exécutera une commande. La deuxième ligne définie la commande qui sera exécutée : c'est le script `/bin/traitement-notification`. C'est donc ce script (que nous écrirons par la suite) qui va traiter les notifications reçues. Le mot `default` signifie que le programme sera appelé quelle que soit la notification reçue. On peut le remplacer par un OID si l'on souhaite appeler ce programme uniquement dans le cas de la réception d'une notification ayant un OID spécifique.

À chaque invocation d'un script de traitement de notifications, `snmptrapd` lui envoie sur son entrée standard :
— une 1ère ligne contenant le nom de l'équipement source de la notification ;
— une 2ème ligne contenant les IP et ports source et destination ;
— et, pour chaque objet contenu dans la notification, une ligne avec son OID et sa valeur.

Pour tester, on partira du script minimal ci-dessous :

```
#!/bin/bash
read nom
echo "Nom : "$nom
read ip
echo "IPs et ports : "$ip
while read obj ; do
    echo "Objet : "$obj
done
```

I 4.2 Créez le script `/bin/traitement-notification` avec ce contenu.
I 4.3 Arrêtez le service `snmptrapd`.
I 4.4 Lancer `snmptrapd` dans un terminal (sans passer par systemctl) avec l'option `-f`. Le service ne vous rend pas la main. (On le lance de cette façon afin de voir plus facilement ce qu'il affiche. Sinon il faudrait regarder dans le journal.)
I 4.5 Générez l'envoi d'une notification par l'agent pour vérifier que les informations sont bien affichées par le script.
I 4.6 Arrêtez le processus `snmptrapd`. (Dans la suite, vous pourrez à nouveau utiliser `systemctl` pour lancer `snmptrapd`.)

L'objectif final est d'écrire un script de traitement des notifications permettant d'envoyer un mail à l'administrateur suite à la réception d'une notification. Ce mail devra contenir :
— la date de réception de la notification ;
— l'adresse IP et le nom de l'équipement ayant envoyé la notification ;
— et l'OID de la notification envoyée.

I 4.7 Modifiez le fichier `traitement-notification` en conséquence puis tester. Pour l'adresse mail, utiliser un service d'adresses mail jetables comme, par exemple, `https://getnada.com/`.