TP2 : Surveillance de la MIB et notifications SNMP

Ce TP est à réaliser en binôme. Chaque binôme dispose de deux machines Linux : une machine NMS (manager) et une machine à superviser (agent).

1. Prérequis

1. Sur les deux machines (Agent et NMS), installer les paquets suivants :

   snmp, snmp-mibs-downloader et **wireshark**.

2. Sur la machine Agent : installer le paquet **snmpd**.

3. Sur le NMS : installer les paquets **snmptrapd** (serveur de traitement des notifications).

4. Vérifier que vous avez **postfix** (serveur d'envoi de mails) bien installé et configuré sur votre machine NMS (revoir le TP1 pour plus d'informations).

5. Sur les deux postes Agent et NMS : commenter la ligne **mibs** dans le fichier /etc/snmp/snmp.conf (en ajoutant un # en début de ligne).

   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).

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

   Sur la machine Agent, supprimer puis recréer ce fichier avec le contenu minimal ci-dessous :

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

   N'oubliez pas de redémarrer le service (systemctl restart snmpd) après chaque modification de ce fichier.

2. Surveillance des processus

Il est possible avec les outils Net-SNMP de déclencher l'envoi de notifications à la suite de 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). C'est une MIB privée dont l'OID est 1.3.6.1.4.1.2021.
Vous pouvez consulter la MIB UCD à travers ce lien :
https://cric.grenoble.cnrs.fr/Administrateurs/Outils/MIBS/?module=UCD-SNMP-MIB&fournisseur=netsnmp

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

1. Nous allons maintenant configurer l'agent pour qu'il envoie des notifications (ou traps) au NMS. Définir, dans le fichier de configuration de l'agent, une communauté pour l'envoi de notifications ainsi que la destination de ces notifications :

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

   Pour la communauté, vous choisirez un nom quelconque.

   La première indique la communauté utilisée pour envoyer des notifications.
   La deuxième ligne indique que toutes les notifications seront au format v2c et envoyées à l'IP précisée.
   Une fois cette modification effectuée, l'agent enverra automatiquement un certain nombre de notifications génériques (p.ex., suite à l'arrêt de l'agent) au NMS.

2. Ajouter 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 des objets de la MIB.

3. Remplacer dans ce fichier la ligne Environment="MIBS=" par Environment="MIBS=ALL" et, dans la ligne ExecStart=..., enlever le paramètre -I -smux,mteTrigger,mteTriggerConf.

4. Exécuter la commande ci-dessous pour que les modifications soient prises en compte :

```
# systemctl daemon-reload
```

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

5. Ajouter 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.

6. Démarrer le service **sshd** s'il n'est pas lancé.

7. Récupérer, 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 ».

8. Ajouter 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.

9. Tout en capturant les trames, arrêter le serveur sshd sur l'agent. On devrait normalement capturer la notification. Prener une copie d'écran du contenu de la notification capturée.

   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).

10. Redémarrer le service **sshd**.

11. En consultant le manuel de snmpd.conf, trouver comment préciser que l'on souhaite toujours avoir un et un seul processus **sshd**.

12. Vérifier qu'une notification est bien envoyée lorsque l'on ouvre une 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.

    Donner une copie d'écran du contenu de la notification capturée.

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.

1. Ajouter 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 communiquant des informations sur le fichier observé ont été créés dans la MIB.

2. Vérifier, avec **snmpwalk** que la branche 1.3.6.1.4.1.2021.15 est bien présente avec les informations adéquates.

3. En vous inspirant de ce qui a été fait au point 8 de la section précédente et en observant le résultat de la commande **snmpwalk** utilisée au point précédent, faire 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, utiliser un OID de notification inexistante.

4. Tester et vérifier avec **wireshark** que la notification a bien été envoyée.

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.

1. Sur le NMS, modifier le fichier de configuration de snmptrapd pour qu'il contienne uniquement la ligne ci-dessous :

   authCommunity log,execute <communaute-de-notifications >

   Cette ligne signifie qu'à chaque notification provenant de la communauté précisée, le NMS effectuera deux opérations :
   — log => enregistrement de la notification reçue dans le journal
   — execute => exécution d'une commande (voir points suivants)
   (Une autre opération est possible : net. Elle permet de rediriger la notification vers une autre machine.)

2. À l'aide de la commande **snmptrap**, envoyer une notification depuis l'agent pour vérifier qu'elle est bien enregistrée dans le journal (fichier /var/log/syslog).

3. Ajouter la ligne ci-dessous dans le fichier snmptrapd.conf :

   traphandle default /bin/traitement-notification

   La ligne ajoutée signifie que le service snmptrapd exécutera la commande */bin/traitement-notification* à chaque fois qu'il recevra une 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 IPs 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
```

4. Créer le script /bin/traitement-notification avec ce contenu.

5. Arrêter le service snmptrapd.

6. 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.)

7. Générer l'envoi d'une notification par l'agent pour vérifier que les informations sont bien affichées par le script.

8. Arrêter le processus snmptrapd. (Dans la suite, vous pourrez à nouveau utiliser systemctl pour lancer snmptrapd.)

   Le but de cet exercice 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.

9. Modifier 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/.

BONUS

10. Modifier le script pour que l'adresse mail de l'administrateur ne soit plus fixée dans le script, mais récupérée depuis la MIB de l'agent (objet sysContact auquel vous aurez donné pour valeur une adresse électronique de test).

11. Modifier le script pour que le mail contienne également les différents objets contenus dans la notification.