Ingénieurs 2000 - SNMP

Utilisation de SNMP

VIVIEN BOISTUAUD
LAURENT BOIVIN

Université de Marne la vallée - UFR Ingénieurs 2000
Informatique et Réseaux – 2ème année
Année Universitaire 2006-2007

Table des matières

Introduction

Dans le cadre de nos études d'ingénieurs en informatique et réseaux à l'université de Marne-la-vallée, nous étudions de nombreuses technologies liées au développement d'applications mais également à la gestion et la configuration de réseau.

Parmi les enseignements liés aux réseaux, on peut citer la configuration de firewall et la création de zones démilitarisées, les technologies de réseaux sans fils, les technologies de routage (RIP, OSPF, BGP…) ainsi que le monitoring de réseaux à l'aide de SNMP.

Ce rapport se concentre sur la configuration et l'utilisation du protocole SNMP (Simple Network Management Protocol), qui est un protocole de monitoring et de configuration réseau à distance. De nombreux matériels et systèmes d'exploitation incluent la prise en charge de SNMP, la version dépendant des choix des concepteurs.

Ce rapport est orienté pour une utilisation sous Linux, en conjonction avec du matériel de routage Cisco implantant SNMP. Lorsque cela est possible, les correspondances entre les outils et commandes utilisables sous Microsoft Windows seront fournies.

L'objectif est de permettre au lecteur d'acquérir des connaissances sur le protocole SNMP, l'organisation des informations dans les MIB (Management Information Base) et la configuration et la gestion d'un routeur Cisco par le protocole SNMP. Il peut être utilisé comme aide mémoire pour les administrateurs système souhaitant acquérir un apprentissage basique du protocole SNMP, en complément des documents cités en référence.

1 Matériel utilisé et références utiles

1.1 Configuration utilisée

Lors de ce rapport, nous avons utilisé le matériel suivant :

- Ordinateur Intel P4 fonctionnant sous Debian GNU/Linux v3.1 – noyau 2.6
- Routeur Cisco 2500 Series, équipé de l'IOS 11 de Cisco
- Convertisseur port Ethernet Cisco/RJ-45
- Un câble RJ-45 croisé pour la connexion Ethernet au routeur
- Connecteur pour port série RS332 / serial routeur Cisco

Nous avons également utilisé les logiciels suivants :

- IOS 11 pré installé sur le routeur Cisco 2500 Series, qui inclus la prise en charge du monitoring et du contrôle à distance par SNMP, ainsi que les MIB propriétaires Cisco (disponibles sur : http://monge.univ-mlv.fr/~roussel/SNMP/)
- Le logiciel Net-SNMP client, incluant les commandes `snmpget`, `snmpgetnext`, `snmpset`.
- Le démon Net-SNMP qui permet la gestion des traps et le monitoring distant de la machine linux sur laquelle il est installé vi, respectivement, `trapd` et `snmpd`.
- Le navigateur graphique de MIB (Management Information Base – Base d'information sur la gestion) nommé `mbrowse`.

La configuration utilisée est la suivante :

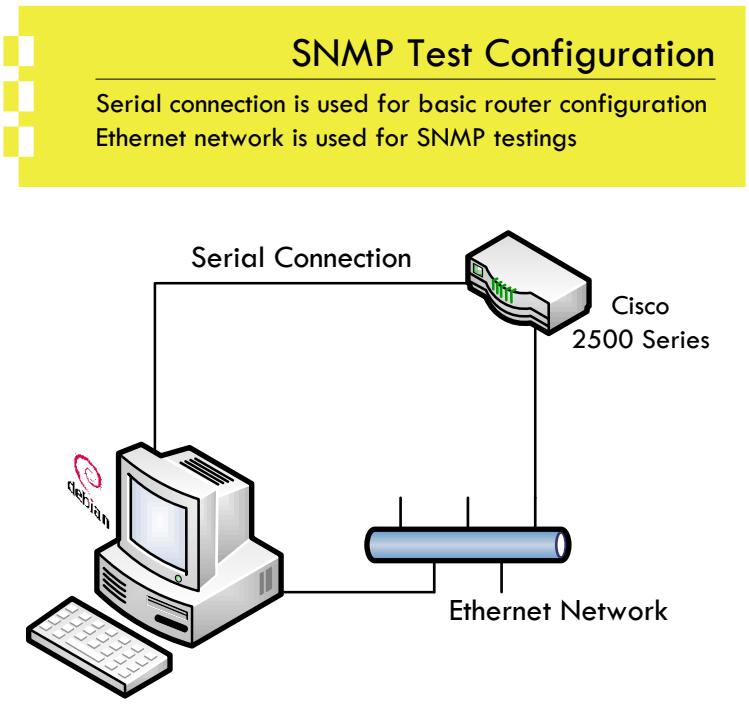

Figure 1 - SNMP Test Configuration used in this document

1.2 Variantes pour les utilisateurs de Windows

Sous Windows, il n'existe pas de logiciel client fourni avec le système pour prendre en charge la consultation et le contrôle distant de périphériques utilisant un agent SNMP. Voici un des nombreux outils tierce partie disponibles pour combler ce manque :

- Getif v2.3 ou supérieur remplace le client Net-SNMP
- La collection de MIB SNMP4tPC indispensable pour Getif
- Un excellent tutoriel, en anglais, sur l'outil Getif

En revanche, les ordinateurs Windows NT à partir de la version 5 (formellement appelée Windows 2000) peuvent être surveillés et contrôlés à distance à l'aide d'un pilote SNMP WMI qui est fourni avec le système mais non installé par défaut.

Pour l'installer sous Windows 2000/XP/2003, ouvrez le panneau de configuration, sélectionnez « Ajout/Suppression de programmes » puis sélectionnez sur la gauche le bouton « Ajouter/Supprimer des composants Windows ». Dans la liste des composants, sélectionnez la ligne « Outil de gestion et d'analyse » puis cliquez sur le bouton « Détails ». Cochez ensuite la case correspondante à « SNMP (Protocole simplifié de gestion de réseau) » et validez les deux fenêtres ouvertes en appuyant sur les boutons OK correspondants.

Pour installer l'agent SNMP sous Windows Vista (Edition professionnelle, entreprise ou intégrale), ouvrez le panneau de configuration et sélectionnez l'icône « Programme et fonctionnalités » (si vous êtes dans l'affichage par regroupements, cette option se situe dans le sous menu « Programmes »). Sélectionnez alors sur la droite l'option « Activez ou Désactivez des fonctionnalités Windows » et dans le dossier « Fonctionnalités SNMP », sélectionnez l'option « Fournisseur SNMP WMI ». Notez que la version prise en charge est SNMPv2c, qui est une version compatible avec SNMPv1.

Veuillez également noter que les administrateurs Windows les plus avertis pourront configurer WMI (Windows Management Interface) pour convertir les traps SNMP en notifications WMI, ou pour consulter et configurer à distance des services SNMPv1/v2c via WMI au lieu d'utiliser le client Getif. Cependant, ces procédures ne seront pas détaillées dans ce document. Pour plus d'informations référez vous au site de Microsoft.

1.3 SNMP et les communications réseaux

Le protocole SNMP utilise le protocole de transport UDP/IP, notamment le port 161 (pour la communication entre service de monitoring et l'agent SNMP) et les ports 162 (pour l'envoi des traps SNMP – notification d'évènement).

Aussi, il appartient à l'utilisateur de s'assurer que le port UDP 161 peut être ouvert et ne sera pas bloqué sur les machines exécutant un agent SNMP (les machines devant être configurées via SNMP).

Il appartient également à l'utilisateur de s'assurer que la machine utilisée pour le contrôle du réseau par SNMP, et désignée pour recevoir des traps, peut utiliser le port UDP 162.

Notez que sous linux, seul une application s'exécutant avec le compte administrateur (`root`) peut ouvrir les ports TCP et UDP compris entre 0 et 1023 ; ceci pour des raisons de sécurité.

1.4 Références et liens utiles

Ces références pourront être utiles au lecteur afin de mieux comprendre les principes de fonctionnement du protocole SNMP :

- Cours sur le protocole SNMP par Gilles Roussel – Université de Marne-la-vallée (http://monge.univ-mlv.fr/~roussel/SNMP/SNMP.pdf)
- RFCs définissant le protocole SNMP v1 – RFC 1555 – RFC 1556 – RFC 1557
- De nombreuses autres RFC définissent les protocoles SNMP v2 et v3, si vous souhaitez obtenir leur référence, reportez vous à l'excellente présentation de Gilles Roussel.
- Manuel d'utilisation des routeurs Cisco de la série 2500
- Manuel des commandes de l'IOS 12 (Internet Operating System v12) de Cisco

De nombreuses autres documentations sont disponibles sur internet, notamment sur les sites de l'IETF (*Internet Engineering Task Force*) et sur le site de Wikipedia.

2 Configuration du client SNMP

2.1 Pré-requis

Avant de pouvoir utiliser les outils SNMP, il est nécessaire de les installer. Pour les utilisateurs de Windows, reportez-vous à la section 1.2. Pour les utilisateurs de Debian GNU/Linux, vous pouvez utiliser le gestionnaire de paquets APT pour installer les outils nécessaires à l'aide de la commande suivante :

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

Remarque : Dans la suite de ce document, les commandes destinées au poste linux et précédées d'un # (dièse – sharp) symbolisent une commande à exécuter en tant qu'administrateur. Les commandes précédées d'un $ (dollar) peuvent en revanche être exécutées en tant que simple utilisateur du système.

2.2 Configuration souhaitée

On souhaite configurer notre client SNMP pour qu'il utilise la version `1` du protocole (version utilisée par les routeurs Cisco 2500), qu'il soit rattaché à la communauté SNMP nommée `ig2k`, que le chemin où sont stockées les descriptions de MIB et de SMI soit `/usr/share/snmp/mibs:/tmp/mibs/`, et que toutes les MIB et SMI soient chargées par défaut (option `+ALL`).

2.3 Génération de la configuration du client SNMP

Il existe trois fichiers de configuration SNMP sous linux, à savoir :

- `snmp.conf` qui est utilisé pour la configuration du client SNMP,
- `snmpd.conf` qui est utilisé pour la configuration du serveur SNMP (aussi appelé agent SNMP),
- `snmptrapd.conf` qui est utilisé pour configurer le service de gestion des traps émis par d'autres agents SNMP à destination d'une machine superviseur.

Il est possible d'obtenir plus d'informations sur la syntaxe de ces différents fichiers de configuration et les commandes qui y sont autorisées en consultant les pages de manuel correspondantes :

```
$ man snmp.conf snmpd.conf snmptrapd.conf
```

Cependant, il existe également un outil qui permet de générer ces fichiers de configuration à l'aide d'un assistant utilisable en mode console. Cet outil se nomme `snmpconf` et peut être utilisé sans être administrateur.

Les fichiers de configuration générés sont stockés dans le répertoire courant par défaut : vos fichiers de configuration actuels ne seront pas écrasés par cet assistant, à moins que vous ne l'exécutiez en tant qu'administrateur dans le répertoire `/etc/snmp` de votre système, ce qui est déconseillé.

Pour lancer l'outil de configuration SNMP, on utilise donc :

```
$ snmpconf
```

Si des fichiers de configuration existent déjà, vous obtiendrez le message suivant :

```
The following installed configuration files were found:

   1:  /etc/snmp/snmpd.conf
   2:  /etc/snmp/snmptrapd.conf

Would you like me to read them in?  Their content will be merged with the
output files created by this session.

Valid answer examples: "all", "none","3","1,2,5"

Read in which (default = all): none
```

Entrez alors `none` comme réponse, afin que la configuration actuelle ne soit pas prise en compte et ne perturbe notre configuration. Il vous sera ensuite demandé de sélectionner le fichier que vous souhaitez éditer :

```
I can create the following types of configuration files for you.
Select the file type you wish to create:
(you can create more than one as you run this program)

   1:  snmpd.conf
   2:  snmptrapd.conf
   3:  snmp.conf

Other options: quit

Select File: 3
```

Remarque générale : les numéros des options et leur ordre peut changer suivant la version de `snmpconf` que vous utilisez, notamment sur les dernières versions de Debian Etch, qui diffère de la version Sarge utilisée lors de la rédaction de ce rapport.

Choisissez alors `snmp.conf` (ici l'option `3`). Une liste des éléments configurables, regroupés par thèmes, est alors affichée :

```
The configuration information which can be put into snmp.conf is divided
 into sections.  Select a configuration section for snmp.conf
 that you wish to create:

   1:  Debugging output options
   2:  Default Authentication Options
   3:  Output style options
   4:  Textual mib parsing

Other options: finished

Select section:
```

On choisit alors l'option `Default Authentication Options` (ici l'option `2`). On obtient alors le menu suivant :

```
Section: Default Authentication Options
Description:
  This section defines the default authentication
  information.  Setting these up properly in your
  ~/.snmp/snmp.conf file will greatly reduce the amount of
  command line arguments you need to type (especially for snmpv3).

Select from:

   1:  The default port number to use
   2:  The default snmp version number to use.
   3:  The default snmpv1 and snmpv2c community name to use when needed.
   4:  The default snmpv3 security name to use when using snmpv3
   5:  The default snmpv3 context name to use
   6:  The default snmpv3 security level to use
   7:  The default snmpv3 authentication type name to use
   8:  The default snmpv3 authentication pass phrase to use
   9:  The default snmpv3 privacy (encryption) type name to use
  10:  The default snmpv3 privacy pass phrase to use

Other options: finished, list

Select section:
```

On choisi alors l'option toto `The default snmp version number to use` (ici l'option `2`). On entre alors la version `1` à l'invite suivant. En SNMP version 1, la sécurité se base sur le partage d'un nom de communauté SNMP à utiliser. La configuration de ce nom se fait via l'option `The default snmpv1 and snmpv2c community name to use when needed` (ici l'option `3`). A l'invite suivant on entre alors `ig2k` comme nom de communauté.

On retourne ensuite au menu principal de configuration de `snmp.conf` en entrant le choix `finished`. On doit ensuite configurer les emplacements des fichiers de MIB (Management Information Base) qui définit les hiérarchies utilisées par certains types de périphériques implantant SNMP, afin de convertir des chemins de navigation textuelle en des chemins numériques, utilisés dans les paquets SNMP.

Les MIB dépendent des systèmes avec lesquels on souhaite communiquer. Par exemple pour consulter les informations propriétaires d'un routeur Cisco, il faut installer les MIB Cisco dans la base de données locale du client. Un certain nombre de MIB sont communes à tous les systèmes, ceux-ci étant fournis avec l'application Net-SNMP sous linux (ou disponible en téléchargement pour Windows).

Dans un premier temps, nous utiliseront les MIB standard fournies avec Net-SNMP, qui sont localisée dans le répertoire `/usr/share/snmp/mibs`, puis nous ajouterons par la suite des MIBs supplémentaires dans le répertoire `/tmp/mibs`. De plus, nous configurerons l'application cliente pour qu'elle charge par défaut tous les MIBs disponibles.

Pour ce faire, nous choisissons l'item `Textual mib parsing` (ici l'option `4`). Nous obtenons alors le sous menu suivant :

```
Section: Textual mib parsing
Description:
  This section controls the textual mib parser.  Textual
  mibs are parsed in order to convert OIDs, enumerated
  lists, and ... to and from textual representations
  and numerical representations.

Select from:

   1:  Specifies directories to be searched for mibs.
   2:  Specifies a list of mibs to be searched for and loaded.
   3:  Loads a particular mib file from a particualar path
   4:  Should errors in mibs be displayed when the mibs are loaded
   5:  Should warnings about mibs be displayed when the mibs are loaded
   6:  Be strict about about mib comment termination.
   7:  Should underlines be allowed in mib symbols (illegal)
   8:  Force replacement of older mibs with known updated ones

Other options: finished, list

Select section:
```

On sélectionne alors l'option `Specifies directories to be searched for mibs` (ici l'option `1`). A l'invite suivant on entre alors le texte `/usr/share/snmp/mibs:/tmp/mibs/`, puis on sélectionne l'option `Specifies a list of mibs to be searched for and loaded` (ici l'option `2`). On saisit alors l'option `+ALL`

pour indiquer que toutes les MIBs et SMI présentes dans ces répertoires doivent être chargés.

On valide alors les menu (option `finished`) jusqu'à quitter l'application. On obtient alors un fichier de configuration `snmp.conf` dans le répertoire courant, ressemblant au fichier suivant :

```
# Security Section
defversion    1
defcommunity  ig2k

# MIBs and SMIs to be loaded
mibs          /usr/share/snmp/mibs:/tmp/mibs/
mibs          +ALL
```

Il suffit ensuite de copier ce fichier de configuration dans le répertoire `/etc/snmp` :

```
# cp snmp.conf /etc/snmp/
```

Notre client SNMP est désormais correctement configuré et peut être utilisé avec les commandes `snmpget`, `snmpgetnext`, `snmpset` et `snmptrap`. Celles-ci permettent respectivement de récupérer une information (paquet SNMP GET), récupérer une information faisant partie d'une série (paquet SNMP GET NEXT), modifier une valeur (paquet SNMP SET), ou simuler l'envoi d'un trap SNMP (paquet SNMP TRAP).

3 Configuration initiale du routeur Cisco 2500 series

3.1 Pré-requis

On souhaite maintenant configurer le routeur Cisco 2500 series afin qu'il puisse communiquer avec notre machine par le biais d'une connexion Ethernet. Pour cela, nous allons utiliser une connexion série au routeur, par le biais d'un port RS332 et du logiciel minicom. Si minicom n'est pas installé, utilisez (sous Debian) la commande :

```
# apt-get install minicom
```

Sous Windows, il faut utiliser l'HyperTerminal au lieu de minicom, qui peut être lancé par le menu démarrer, dans le sous menu Accessoires / Communication / HyperTerminal.

La configuration nécessaire pour la communication est :

- Vitesse de 9600 bauds
- Taille de données de 8 bits
- Pas de parité
- 1 bit de stop
- Ne pas utiliser le contrôle de flux matériel ni logiciel

Sous Debian, on lance minicom à l'aide de la commande `minicom`. Pour configurer le port série, il faut appuyer sur les touches `Ctrl-A` puis sur la touche `Z`. Puis, on appuie sur la touche `P` pour accéder au menu de configuration de la connexion.

Puis on sélectionne les options `E`, `L`, `V` et `W`. La ligne « actuellement » doit alors indiquer : `9600 8N1`. La configuration est alors correcte, on ferme alors le menu en appuyant sur entrée, puis une nouvelle fois sur entrée pour déclencher la connexion au routeur.

Vérifiez également que le port de communication auquel le routeur est connecté est bien le port avec lequel `minicom`/HyperTerminal s'attend à communiquer.

Si le routeur ne vous propose pas d'exécuter l'assistant de configuration initiale lors de la connexion, ou s'il n'est pas déjà lancé, alors c'est que sa NVRAM (Non Volatile RAM), qui stocke les informations de configuration, n'est pas vierge. Dans ce cas, utilisez les commandes suivantes pour la supprimer :

```
Router> enable
Router# erase startup-config
Router# reload
```

Si un mot de passe vous est demandé pour devenir super utilisateur (via la commande `enable`), tentez `cisco`, qui devrait être le bon, du moins sur le matériel de l'institut depuis lequel nous avons réalisé ce mode opératoire.

Le routeur devrait alors redémarrer, et vous proposer d'exécuter l'assistant de configuration. Répondez « `no` » à l'exécution de cet assistant car mieux vaut le configurer soit même à la main.

3.2 Configuration de la sécurité du routeur

Par défaut, les routeurs Cisco ne possèdent pas de mot de passe pour accéder au mode super utilisateur via la commande `enable`. Notre première mission va donc consister à en mettre un. Pour cela, passons en mode super utilisateur :

```
Router> enable
Router#
```

Les routeurs Cisco admettent de nombreuses commandes en fonction du mode dans lequel le terminal de configuration se trouve actuellement. Ainsi, lorsque l'invite est :

- NomRouter> c'est que l'accès est en mode utilisateur classique,
- NomRouter# c'est que l'accès est en mode super utilisateur,
- NomRouter(config)# c'est que vous êtes dans le sous mode de configuration
- NomRouter(config-if)# désigne le mode configuration pour une interface réseau (Ethernet, série…)
- NomRouter(config-line)# désigne le mode de configuration pour une interface de terminal virtuel (`vty`…)

Pour passer dans le mode de configuration, une fois administrateur sur le routeur, on tape donc :

```
Router# enable secret
```

Puis on demande à activer le mot de passe (`secret`) `cisco` pour accéder au mode super utilisateur (`enable`), ce qui se fait via la commande :

```
Router(config)# enable secret cisco
Router(config)# ^Z
```

Le mode super utilisateur est désormais protégé par le mot de passe `cisco`. C'est le minimum de sécurité indispensable lors de toute configuration de routeur Cisco suite à une remise à zéro de sa configuration.

Remarque : Le texte `^Z` *ci-dessus signifie qu'il faut appuyer simultanément sur les touches Ctrl et Z de votre clavier. Ce raccourci envoi au routeur un signal qui est interprété par l'IOS comme une demande de retour au mode super utilisateur (invite de commande* `Router#`*). Ce raccourci est utilisable à tout moment à condition d'être dans un sous mode du mode administrateur.*

3.3 Configuration IP du réseau

Afin de pouvoir communiquer entre le routeur et notre machine par un réseau IP (indispensable à l'utilisation de SNMP, qui utilise UDP), nous souhaitons mettre en place la configuration de la figure suivante :

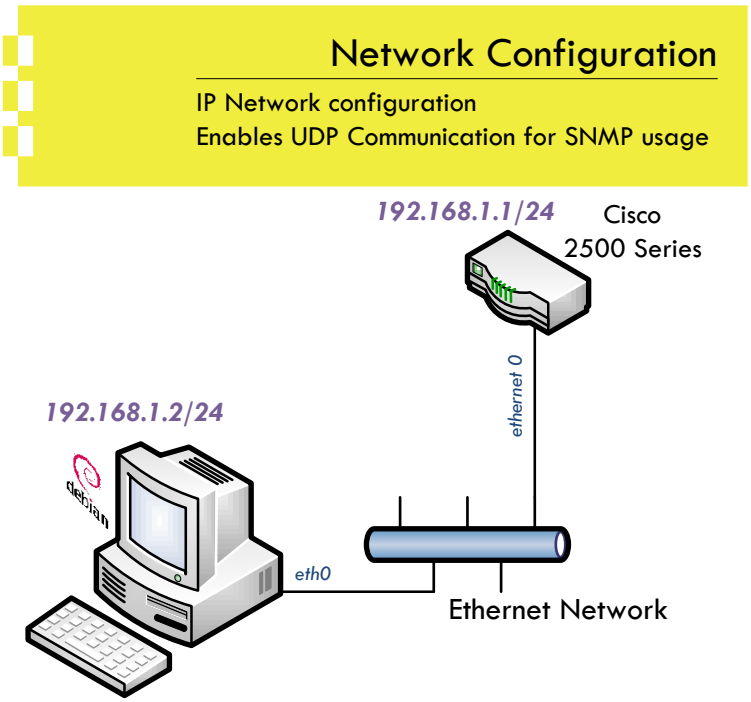

Figure 2 - IP Network configuration used for SNMP testing

3.3.1 Configuration côté routeur

Pour réaliser la configuration IP du réseau telle qu'indiquée ci-dessus, on doit être en mode super administrateur sur le routeur Cisco 2500 et utiliser les commandes suivantes (expliquées ci-après) :

```
Router# configure terminal
Router(config)# interface ethernet 0
Router(config-if)# ip address 192.168.1.1 255.255.255.0
Router(config-if)# no shutdown
Router(config-if)# ^Z
```

La commande `configure terminal` permet de passer le terminal en mode configuration. La commande `interface ethernet 0` (ou `ethernet0`, ou `fastethernet`

`0` suivant les routeurs) permet de passer en mode de configuration de la première interface Ethernet du routeur.

Enfin, la commande `ip address 192.168.1.1 255.255.255.0` permet de définir que l'adresse IP de cette interface est `192.168.1.1` et que le masque de sous réseau est `255.255.255.0` (24 bits à 1). La commande `no shutdown` permet d'activer l'interface réseau.

Remarque générale sur les routeurs Cisco : les commandes dont le nom est précédé de `no` ne doivent pas être prises comme une négation à proprement parler, mais plutôt comme une commande ayant l'effet inverse de celui dont le nom suit le `no`. Ainsi, il ne faut pas interpréter « `no shoutdown` » comme signifiant « ne pas éteindre » (ce qui serait la traduction littérale), mais comme signifiant « faire l'inverse de la commande `shutdown` » ou, en d'autres termes, « allumer l'interface concernée ».

3.3.2 Configuration côté PC Linux

Pour que l'ordinateur que nous utilisons puisse communiquer avec notre routeur Cisco, il est indispensable qu'ils soient sur le même réseau IP, c'est-à-dire 192.168.1.0/24. L'adresse IP que nous souhaitons affecter à notre machine Linux est 192.168.1.2. Pour cela, nous utilisons la commande `ifconfig` comme suit :

```
# ifconfig eth0 inet 192.168.1.2 netmask 255.255.255.0 up
# ifconfig
```

La seconde saisie de la commande `ifconfig` permet de vérifier que l'interface `eth0` a été correctement configurée. Dans le cas contraire, il est possible que votre interface porte un autre nom que `eth0`, remplacez cette information le cas échéant.

On peut vérifier que la communication se fait correctement depuis le PC linux à l'aide de la commande :

```
# ping 192.168.1.1
PING 192.168.1.1 (192.168.1.1) 56(84) bytes of data.
64 bytes from 192.168.1.1: icmp_seq=1 ttl=128 time=0.321 ms
64 bytes from 192.168.1.1: icmp_seq=2 ttl=128 time=0.167 ms
64 bytes from 192.168.1.1: icmp_seq=3 ttl=128 time=0.166 ms
64 bytes from 192.168.1.1: icmp_seq=4 ttl=128 time=0.162 ms
64 bytes from 192.168.1.1: icmp_seq=5 ttl=128 time=0.164 ms
--- 192.168.1.1 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 3999ms
rtt min/avg/max/mdev = 0.162/0.196/0.321/0.062 ms
```

Ou depuis le routeur :

```
Router# ping 192.168.1.2
```

3.4 Configuration du serveur Telnet du routeur

Maintenant que le réseau a été configuré correctement, nous pouvons configurer le routeur pour qu'il puisse être configuré par une connexion Telnet en complément de la connexion série. Ceci permet de configurer à distance le routeur, du moment que son adresse IP est accessible depuis la machine depuis laquelle nous tentons de nous connecter.

L'inconvénient du protocole Telnet est ce que les informations sont envoyées en clair sur la connexion. Des protocoles plus récents, comme SSH, permettent d'obtenir une connexion cryptée mais ce protocole n'est pas implanté sur l'IOS 11 de Cisco.

Malgré cela, à des fins démonstratives, nous allons configurer le service Telnet de notre routeur. Nous utilisons à ces fins les commandes suivantes :

```
Router# configure terminal
Router(config)# line vty 0 4
Router(config-line)# password cisco
Router(config-line)# login
```

La commande `configure terminal` permet de passer le terminal en mode configuration. La commande suivante, `line vty 0 4` permet de placer le routeur en mode configuration du terminal virtuel Telnet (nommé en interne `vty 0`) pour un maximum de 5 connexions (`4 + 1`) simultanées.

La commande `password` permet ensuite de configurer le mot de passe nécessaire à la connexion Telnet, tandis que `login` permet d'activer le droit de se loger par Telnet sur le routeur. Le mot de passe n'est pas tenu d'être le même que le mot de passe administrateur, il est d'ailleurs déconseillé que ces deux mots de passe soient les mêmes.

On peut désormais se connecter au routeur par Telnet depuis notre ordinateur à l'aide de la commande :

```
$ telnet 192.168.1.1
```

Une fois la connexion établie, l'utilisation du mot de passe `cisco` permettra de se loger directement et d'administrer le routeur de la même façon que si nous y étions connectés directement par une connexion série.

4 Configuration de l'agent SNMP du routeur

Les routeurs Cisco de la série 2500 et supérieure peuvent être supervisés et configurés à distance en utilisant le protocole SNMP version 1. Afin de pouvoir achever ces tâches, il est nécessaire de configurer l'agent SNMP du routeur pour qu'il puisse réponde aux requêtes SNMP et, éventuellement, qu'il transmette ses avertissements (`traps`) à une machine de supervision.

Cette section nous permet d'entrer dans le vif du sujet en configurant le routeur Cisco pour qu'il puisse être supervisé et administré à distance par le protocole SNMP, actions que nous aborderons dans la section suivante.

4.1 Personnalisation de champs SNMP

Afin de fournir des informations administratives utiles sur le routeur via le protocole SNMP, il est nécessaire de configurer au moins deux champs SNMP qui pourront être interrogés à distance : l'emplacement du routeur (champs location) et l'adresse de la personne à contacter en cas de problème avec ce routeur (champs contact).

Pour cela, nous devons nous mettre dans le mode de configuration du routeur et utiliser les commandes suivantes :

```
Router# configure terminal
Router(config)# snmp-server location IG2K - Salle Reseau 2059
Router(config)# snmp-server contact root@ig2k.fr
```

La commande `snmp-server` permet de configurer les options de l'agent SNMP du routeur. Les commandes `snmp-server location` et `snmp-server contact` permettent donc de modifier les champs `location` (emplacement physique) et `contact` de l'agent SNMP. Ces informations seront consultables à distance par les clients SNMP du réseau appartenant à une communauté autorisée et ayant éventuellement une adresse IP autorisée par les administrateurs du routeur.

4.2 Configuration des communautés et droits d'accès

Pour configurer les communautés et les filtrages de droits d'accès par adresse IP, nous allons utiliser les 2 commandes suivantes de l'IOS :

```
Router(config)# snmp-server community comm_name accs_right [accs_list_grp]
Router(config)# access-list accs_list_grp accs_type ip_addr_or_mask
```

Ces deux commandes permettent respectivement de définir une communauté SNMP de nom `comm_name`, ayant comme droits d'accès `accs_right` (`RW` pour lecture/écriture et `RO` pour lecture seule). Enfin, un paramètre optionnel (`accs_list_grp`) permet d'indiquer un numéro de liste d'accès contenant des règles de filtrage IP restreignant les accès SNMP, plus sûr qu'en se basant uniquement sur le nom de communauté.

La seconde commande, `access-list`, permet de définir une règle à appliquer sur une liste d'accès. Le paramètre `accs_list_grp` précise le numéro de la liste, le paramètre `accs_type` définit le type d'ajout qui est fait (`permit` ou `deny`), et le paramètre `ip_addr_or_mask` permet de définir une ou plusieurs adresses qui seront autorisées ou refusées sur cette liste d'accès : soit par une adresse IP unique (type `192.168.1.2`), soit par un groupement d'adresses IP désignées à l'aide d'un masque (type `192.168.1.0/24`).

Pour nos tests, nous allons configurer deux communautés et une liste d'accès :

- La communauté `ig2k`, qui aura un accès en lecture depuis n'importe quelle adresse IP accessible depuis le routeur,
- La communauté `secret`, qui aura un accès en lecture et en écriture, uniquement depuis l'adresse IP `192.168.1.2` configurée sur la liste d'accès numéro `1`.

Pour cela, nous devons utiliser les commandes suivantes :

```
router(config)#snmp-server community ig2k RO

router(config)#access-list 1 permit 192.168.1.2
router(config)#snmp-server community secret RW 1
```

Ainsi, la configuration est conforme à celle décrite ci-avant.

4.3 Configuration des alarmes (traps) SNMP

Lors de certains évènements importants, comme l'arrêt d'une machine, la mise hors tension d'une carte réseau, l'allumage d'un routeur etc., il est possible de configurer les agents SNMP pour qu'ils transmettent des alertes (appelées `trap`) à un serveur de supervision SNMP.

Ces traps permettent d'assurer une supervision efficace des machines du réseau lorsque celles-ci proposent un agent SNMP. Cependant, dans la mesure où les messages du protocole SNMP sont transportés par UDP, il n'y a aucune garantie de remise des messages et ainsi, SNMP v1 ne permet pas de garantir une information à 100% fiable et en temps réel. Cependant, cela reste très utile.

L'agent SNMP du routeur Cisco 2500 assure une gestion des alarmes qui est désactivée par défaut. Pour cela, nous devons configurer certains paramètres :

- L'adresse du serveur de supervision qui doit recevoir les alarmes (192.168.1.2) ainsi que la communauté pour laquelle l'alarme est envoyée (`ig2k`),
- L'interface réseau qui sera utilisée comme source des alarmes (`ethernet 0`) afin d'éviter d'envoyer les messages sur toutes les interfaces et qu'éventuellement ils soient compromis par des utilisateurs espionnant le réseau,
- Enfin, il faudra indiquer au routeur qu'il doit transmettre toutes les alarmes lorsque les conditions d'envoi se produisent.

Ceci se traduit, en commandes IOS par :

```
Router(config)# snmp-server host 192.168.1.2 ig2k
```

Cette commande permet de désigner la machine ayant pour adresse IP 192.168.1.2 comme étant le superviseur des alarmes SNMP, qui recevra ces alarmes comme provenant de la communauté nommée `ig2k`.

```
Router(config)# snmp-server trap-source ethernet 0
```

Cette commande permet de limiter l'envoi des alarmes SNMP à l'interface `ethernet 0`.

Remarque sur les appellations des interfaces de communication sur les routeurs Cisco : les numéros des interfaces peuvent être collés ou non au type de celle-ci. Par exemple, `ethernet0` *est strictement équivalent à la notation* `ethernet 0` *et toutes les commandes de l'IOS tolèrent ces deux notations.*

```
Router(config)# snmp-server enable traps
```

Cette commande permet d'activer l'ensemble des *traps* SNMP de sorte qu'ils soient transmis au serveur de supervision SNMP configuré précédemment.

La configuration de l'agent SNMP du routeur Cisco est alors terminée. Nous n'aurons plus besoin pour le moment de configurer le routeur Cisco, mais activeront par la suite une interface non utilisée afin de pouvoir tester la modification de configuration par SNMP. Aussi, il est conseillé de garder un terminal (`minicom` ou `telnet`) ouvert pour la suite des nos manipulations.

5 Utilisation des outils SNMP

5.1 Découverte des outils SNMP

Dans un premier temps, nous allons découvrir les outils SNMP en utilisant le démon Net-SNMP local afin de s'assurer d'une prise en charge rapide et sans risque de problèmes liés à une mauvaise configuration réseau. Pour cela, nous utilisons le fichier de configuration `/etc/snmp/snmpd.conf` par défaut, ci-joint en annexe.

La communauté par défaut selon cette configuration s'appelle `public` et elle est en accès en lecture seule. Pour démarrer le démon snmp (qui n'est pas actif par défaut après installation), il faut utiliser le script suivant :

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

Les commandes et applications que nous allons utiliser sont décrites ci-après, ainsi que leur rôle et leur usage.

5.1.1 *snmptranslate*: convertir un chemin SNMP en identifiant MIB

Cet outil permet de convertir un chemin d'accès SNMP en identifiant MIB et vice-versa. Le protocole SNMP utilise une hiérarchie numérique pour l'identification des clés de la *Management Information Base*. Afin de simplifier le travail des administrateurs réseaux, ces chiffres peuvent êtres convertis sous forme textuelle qui représente des noms courts significatifs.

Par exemple, le chemin `.iso.org.dod.internet.mgmt.mib-2.system.sysDescr` représente la description d'un système SNMPv2c. Sa correspondance numérique est `1.3.6.1.2.1.1.1`.

On peut utiliser indifféremment une représentation textuelle ou numérique avec la commande `snmptranslate`. Le commutateur `-Td` permet d'obtenir une description complète de l'objet donc le nom/numéro est passé en paramètre comme suit :

```
$ snmptranslate -Td .iso.org.dod.internet.mgmt.mib-2.system.sysDescr
SNMPv2-MIB::sysDescr
sysDescr OBJECT-TYPE
  -- FROM       SNMPv2-MIB, RFC1213-MIB
  -- TEXTUAL CONVENTION DisplayString
  SYNTAX        OCTET STRING (0..255)
  DISPLAY-HINT  "255a"
  MAX-ACCESS    read-only
  STATUS        current
  DESCRIPTION   "A textual description of the entity.  This value should
                include the full name and version identification of
                the system's hardware type, software operating-system,
                and networking software."
::= { iso(1) org(3) dod(6) internet(1) mgmt(2) mib-2(1) system(1) 1 }
```

Si vous souhaitez uniquement obtenir l'OID (Object Identifier) correspondant à un objet particulier, vous pouvez utiliser le commutateur `-On` comme suit :

```
$ snmptranslate -On .iso.org.dod.internet.mgmt.mib-2.system.sysDescr
.1.3.6.1.2.1.1.1
```

Enfin, il est possible de faire une recherche à partir d'un chemin partiel avec le commutateur `-IR` comme suit :

```
$ snmptranslate -On -IR sysDescr
.1.3.6.1.2.1.1.1
```

5.1.2 *snmpwalk*: naviguer dans la table SNMP d'un système

Cet outil permet de lister les entrées d'une branche de la table SNMP d'un système distant à partir de son adresse IP et de l'identifiant de l'objet (OID textuel ou numérique). La syntaxe est usuellement la suivante :

```
$ snmpwalk ip_address OID
```

Par exemple :

```
$ snmpwalk 192.168.1.1 .1.3.6.1.2.1.4.20.1
IP-MIB::ipAdEntAddr.192.168.1.1 = IpAddress: 192.168.1.1
IP-MIB::ipAdEntIfIndex.192.168.1.1 = INTEGER: 1
…
```

5.1.3 *snmpget* et *snmpgetnext*: consulter une ou plusieurs entrées SNMP

La commande `snmpget` permet de consulter des informations sur un objet consultable d'une entité réseau. Par défaut cette commande utilise le fichier de configuration `snmp.conf` pour déterminer le nom de la communauté ainsi que la version de SNMP à utiliser pour la communication, et tous les paramètres définis précédemment dans la section 2.2. Lorsque l'entité consultée fait partie d'une liste, la commande `snmpget` ne retourne que le premier élément de cette liste.

La commande `snmpgetnext` permet en revanche de récupérer l'entrée suivante d'une liste d'entrées SNMP. Elle utilise les paquets de type SNMP GETNEXT pour cela, et sa configuration se base sur le fichier de configuration `snmp.conf`.

Si vous souhaitez utiliser un autre nom de communauté pour l'utilisation de ces commandes, vous pouvez utiliser le commutateur `-c comm_name` qui permet d'outrepasser le nom de communauté du fichier de configuration.

5.1.4 *snmpset*: pour modifier une valeur par SNMP

La commande `snmpset` permet d'envoyer une requête de type SNMP SET à un matériel réseau autorisant la réception de ce type de message, et uniquement sur des objets (identifiés par leur OID) qui sont modifiables. De plus, les communautés sont en générales restreintes en lecture seule et certaines communautés seulement ont les droits d'accès en écriture.

Ainsi, pour pouvoir communiquer avec notre routeur Cisco en écriture, il faut appartenir à la communauté portant comme nom `secret`, ce qui peut être modifié à l'aide du commutateur `-c comm_name` de la commande `snmpset`.

La syntaxe de cette commande est la suivante :

```
snmpset [-c comm_name] OID TYPE VALUE [OID TYPE VALUE ...]
```

Où `OID` désigne l'identifiant numérique ou textuel de l'objet à modifier, `TYPE` désigne une lettre indiquant le type de valeur à modifier (`i` pour entier, `u` pour un entier non signé, `s` pour une chaîne de caractères…). Enfin, `VALUE` précise la nouvelle valeur à activer comme valeur active pour l'objet modifié.

5.1.5 *mbrowse*: une navigation en mode graphique (X11)

Ce navigateur graphique de MIB SNMP permet d'envoyer des requêtes GET, SET, WALK et de naviguer graphiquement dans la hierarchie MIB du périphérique SNMP étudié. Les champs *host name*, *read community* et *write community* permettent respectivement de configurer l'adresse IP de l'agent SNMP, le nom de communauté à utiliser pour les consultations en lecture et celle utilisée pour les accès en écriture.

On peut ensuite naviguer graphiquement dans la hiérarchie de MIB et envoyer les requêtes souhaitées comme si nous utilisions les commandes `snmpwalk`, `snmpget` et `snmpset`.

Un exemple de fenêtre de `mbrowse` est disponible sur la figure 3 ci-après.

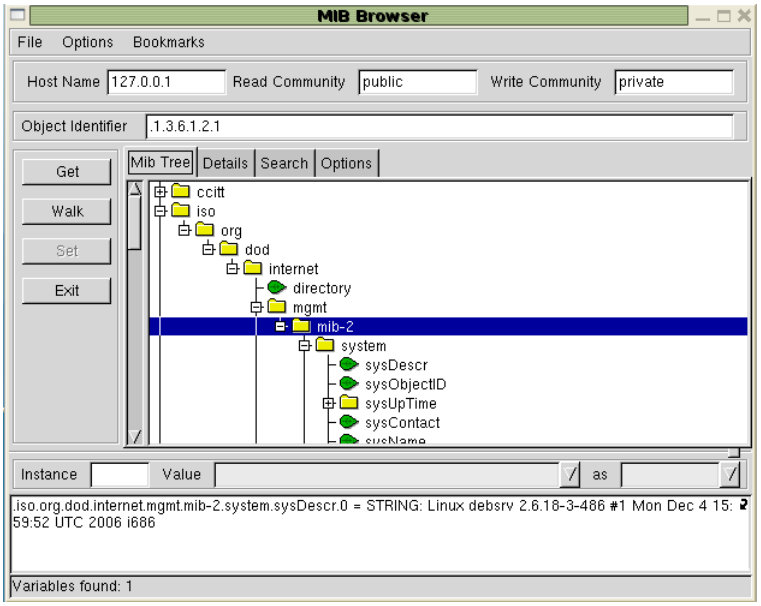

Figure 3 - mbrowse example screenshot

5.1.6 *Ethereal*: un outil de monitoring réseau

`Ethereal` est l'outil graphique de référence en matière de capture de paquets réseau sous Linux. Il permet d'étudier les paquets qui transitent sur un réseau directement connecté à la machine et est un équivalent graphique de `tcpdump` qui propose en plus de puissants outils d'analyse pour de nombreux protocoles.

Il est très utile dans le domaine du réseau et est conseillé dans l'étude des différents protocoles standard, ou non, de l'internet, ainsi que pour détecter les éventuelles intrusions et problèmes réseaux visibles depuis une machine Linux, Mac ou Windows.

Pour plus d'informations, consultez le site d'Etheral.

Remarque pour certains systèmes linux : Si vous installez le paquet ethereal sur une distribution linux récente, il est possible que celui-ci installe en fait l'utilitaire wireshark, qui est une version d'ethereal modifiée et proclamée comme « la nouvelle version d'ethereal ». Nous n'entrerons pas ici dans des débats sur les reprises de projets opensource et nous contenteront de préciser que WireShark peut être utilisé de la même façon qu'ethereal.

5.2 *Récupération d'informations avec les outils*

5.2.1 Informations sur l'objet sysUpTime

Dans un premier temps, nous souhaitons obtenir de nombreuses informations sur l'objet SNMP *sysUpTime* comme :

- Le nom complet de l'objet *sysUpTime*,
- Son numéro d'objet (OID numérique),
- Le fichier de MIB dans lequel se trouvent les informations de cet objet.

Pour cela, nous allons utiliser la commande `snmptranslate`. Pour plus d'informations sur cette commande vous pouvez consulter les pages de manuels de `snmptranslate` et `snmpcmd` (qui contient les informations sur les commutateurs communs aux commandes SNMP).

Obtention du chemin complet de l'objet

Afin d'obtenir le chemin complet de l'objet, il est nécessaire d'utiliser le commutateur `-Of` (*full path output display* - affichage de sortie du chemin complet) ainsi que le commutateur `-IR` (*random access lookup* – recherche en accès aléatoire) qui permet de rechercher une entrée à partir de son nom partiel.

Ainsi, à l'aide de la commande suivante nous obtenons le chemin complet de l'objet *sysUpTime* dans la MIB :

```
$ snmptranslate -IR -Of sysUpTime
.iso.org.dod.internet.mgmt.mib-2.system.sysUpTime
```

Le chemin d'accès complet de l'objet recherché dans la MIB est donc : `.iso.org.dod.internet.mgmt.mib-2.system.sysUpTime`

Obtention de l'identifiant numérique de l'objet

Le commutateur `-On` (*Output as numeric value* – Sortie en format numérique) de cette même commande permet également d'obtenir l'identifiant numérique d'un objet à partir de son chemin complet (ou partiel si on le conjugue avec le script `-IR`). Ainsi nous obtenons :

```
$ snmptranslate -On .iso.org.dod.internet.mgmt.mib-2.system.sysUpTime
.1.3.6.1.2.1.1.3
```

L'identifiant numérique (OID) de l'objet *sysUpTime* est donc : `1.3.6.1.2.1.1.3`.

Informations complémentaires de l'objet dans la MIB

Le commutateur `-Td` (*details*) de la commande `snmptranslate` permet d'afficher toutes les informations stockées dans la MIB concernant cet objet :

```
$ snmptranslate -Td .iso.org.dod.internet.mgmt.mib-2.system.sysUpTime
SNMPv2-MIB::sysUpTime
sysUpTime OBJECT-TYPE
  -- FROM       SNMPv2-MIB, RFC1213-MIB
  SYNTAX        TimeTicks
  MAX-ACCESS    read-only
  STATUS        current
  DESCRIPTION   "The time (in hundredths of a second) since the
                network management portion of the system was last
                re-initialized."
::= { iso(1) org(3) dod(6) internet(1) mgmt(2) mib-2(1) system(1) 3 }
```

Les informations retournées par cette commande sont les suivantes :

- `FROM` : les fichiers dans lesquels cette entrée est déclarée,
- `SYNTAX` : le nom du format des données stockées dans ce type d'objet, qui est une valeur définie dans le protocole SNMP (`INTEGER`, `UNSIGNED`, `STRING`, `TIMETICKS`...)
- `MAX-ACCESS` : le niveau maximal d'accès applicable sur cet objet, qui est `read-only` si l'objet n'est jamais modifiable, et `read-write` si l'objet est modifiable
- `STATUS` : le statut de la valeur, permettant de savoir si la valeur est mise à jour en temps réel ou si elle reflète un état passé dans le temps
- `DESCRIPTION` : la description textuelle de l'objet, en anglais, dans un format lisible humainement. Sert d'aide mémoire pour les administrateurs systèmes.
- Enfin, la dernière ligne précise l'emplacement de l'objet dans la hiérarchie numérique de la MIB.

Ici, l'objet *sysUpTime* est accessible en lecture seule uniquement, il représente une durée de temps (`TIMETICKS`) et représente le temps depuis lequel le gestionnaire de réseau du périphérique n'a pas été réinitialisé. Sur des systèmes connectés au réseau en permanence, ceci est sensiblement équivalent au temps depuis lequel la machine est en route (aussi appelé *uptime*).

Enfin, le champ `FROM` nous permet de savoir que cet objet est défini à deux endroits dans les fichiers de MIB :

- `/usr/share/snmp/mibs/SNMPv2-MIB.txt` et,
- `/usr/share/snmp/mibs/RFC1213-MIB.txt`.

Le premier fichier correspond aux définitions de MIBs standards du protocole SNMP version 2c, tandis que le second correspond aux MIBs définies dans la RFC 1213, qui définit le protocole SNMP version 1.

Dans notre cas, c'est la définition de la MIB de la RFC 1213 qui nous intéresse comme nous travaillons en SNMPv1. Cependant, comme la version v2c de SNMP est compatible avec la version 1, ces définitions peuvent cohabiter sans conflit.

Les informations sont stockées dans le fichier RFC1213-MIB.txt dans le format suivant :

```
sysUpTime OBJECT-TYPE
    SYNTAX  TimeTicks
    ACCESS  read-only
    STATUS  mandatory
    DESCRIPTION
            "The time (in hundredths of a second) since the
            network management portion of the system was last
            re-initialized."
    ::= { system 3 }
```

Ce qui est sensiblement équivalent à celui décrit précédemment, à l'exception que les informations sont hiérarchisées dans le fichier et le chemin donné n'est donc que partiel.

5.2.2 Aperçu global de l'arborescence de la MIB du routeur

La commande `snmpwalk` permet de naviguer dans l'arborescence des MIBs connues d'un matériel réseau. Elle utilise par défaut la configuration du fichier `/etc/snmp/snmp.conf` pour le nom de la communauté à utiliser et les autres informations de configuration du client Net-SNMP.

A l'aide de la commande suivante, on obtient donc les informations ci-après (raccourcies partiellement à des fins d'exemples) :

```
$ snmpwalk 192.168.1.1
SNMPv2-MIB::sysDescr.0 = STRING: Cisco Internetwork Operating System
Software
IOS (tm) 3000 Software (IGS-I-L), Version 11.1(24a), RELEASE SOFTWARE (fc1)
Copyright (c) 1986-2001 by cisco Systems, Inc.
Compiled Fri 09-Mar-01 19:43 by pnicosia
SNMPv2-MIB::sysObjectID.0 = OID: SNMPv2-SMI::enterprises.9.1.29
SNMPv2-MIB::sysUpTime.0 = Timeticks: (214052) 0:35:40.52
SNMPv2-MIB::sysContact.0 = STRING: root@ig2k.fr
SNMPv2-MIB::sysName.0 = STRING: Router
SNMPv2-MIB::sysLocation.0 = STRING: Ig2k - Salle Reseau 2059
SNMPv2-MIB::sysServices.0 = INTEGER: 6
IF-MIB::ifNumber.0 = INTEGER: 4
IF-MIB::ifIndex.1 = INTEGER: 1
IF-MIB::ifIndex.2 = INTEGER: 2
IF-MIB::ifIndex.3 = INTEGER: 3
IF-MIB::ifIndex.4 = INTEGER: 4
IF-MIB::ifDescr.1 = STRING: Ethernet0
IF-MIB::ifDescr.2 = STRING: Serial0
IF-MIB::ifDescr.3 = STRING: Serial1
IF-MIB::ifDescr.4 = STRING: TokenRing0
IF-MIB::ifType.1 = INTEGER: ethernetCsmacd(6)
IF-MIB::ifType.2 = INTEGER: propPointToPointSerial(22)
IF-MIB::ifType.3 = INTEGER: propPointToPointSerial(22)
```

```
IF-MIB::ifType.4 = INTEGER: iso88025TokenRing(9)
IF-MIB::ifMtu.1 = INTEGER: 1500
IF-MIB::ifMtu.2 = INTEGER: 1500
IF-MIB::ifMtu.3 = INTEGER: 1500
IF-MIB::ifMtu.4 = INTEGER: 4464
IF-MIB::ifSpeed.1 = Gauge32: 10000000
IF-MIB::ifSpeed.2 = Gauge32: 1544000
IF-MIB::ifSpeed.3 = Gauge32: 1544000
IF-MIB::ifSpeed.4 = Gauge32: 16000000
IF-MIB::ifPhysAddress.1 = STRING: 0:0:c:4a:d8:91
IF-MIB::ifPhysAddress.2 = STRING:
IF-MIB::ifPhysAddress.3 = STRING:
IF-MIB::ifPhysAddress.4 = STRING: 0:0:30:52:1b:9
IF-MIB::ifAdminStatus.1 = INTEGER: up(1)
IF-MIB::ifAdminStatus.2 = INTEGER: down(2)
IF-MIB::ifAdminStatus.3 = INTEGER: down(2)
IF-MIB::ifAdminStatus.4 = INTEGER: down(2)
IF-MIB::ifOperStatus.1 = INTEGER: up(1)
IF-MIB::ifOperStatus.2 = INTEGER: down(2)
IF-MIB::ifOperStatus.3 = INTEGER: down(2)
IF-MIB::ifOperStatus.4 = INTEGER: down(2)
IF-MIB::ifLastChange.1 = Timeticks: (21544) 0:03:35.44
IF-MIB::ifLastChange.2 = Timeticks: (6136) 0:01:01.36
IF-MIB::ifLastChange.3 = Timeticks: (6136) 0:01:01.36
IF-MIB::ifLastChange.4 = Timeticks: (6139) 0:01:01.39
IF-MIB::ifInOctets.1 = Counter32: 56127
IF-MIB::ifInOctets.2 = Counter32: 0
IF-MIB::ifInOctets.3 = Counter32: 0
…
```

Ces informations sont le reflet de toutes les informations contenues dans les éléments de la MIB du routeur actuellement connues par notre client. On peut également restreindre l'affichage des informations obtenues par `snmpwalk` en précisant une sous-partie de la hiérarchie que l'on souhaite afficher, par exemple :

```
$ snmpwalk 192.168.1.1 system
SNMPv2-MIB::sysDescr.0 = STRING: Cisco Internetwork Operating System
Software ^M
IOS (tm) 3000 Software (IGS-I-L), Version 11.1(24a), RELEASE SOFTWARE
(fc1)^M
Copyright (c) 1986-2001 by cisco Systems, Inc.^M
Compiled Fri 09-Mar-01 19:43 by pnicosia
SNMPv2-MIB::sysObjectID.0 = OID: SNMPv2-SMI::enterprises.9.1.29
SNMPv2-MIB::sysUpTime.0 = Timeticks: (218512) 0:36:25.12
SNMPv2-MIB::sysContact.0 = STRING: root@ig2k.fr
SNMPv2-MIB::sysName.0 = STRING: Router
SNMPv2-MIB::sysLocation.0 = STRING: Ig2k - Salle Reseau 2059
SNMPv2-MIB::sysServices.0 = INTEGER: 6
```

Nous obtenons ainsi les informations relatives au système (au nœud `system` de la MIB du routeur Cisco 2500 series), dont la valeur de l'objet *sysUpTime*.

5.2.3 Récupération de la valeur SysUpTime

On peut obtenir la valeur *sysUpTime* en utilisant, bien évidemment, `snmpwalk`, mais les informations obtenues concernent plus que cette simple entrée. De plus, la commande précédente envoi de nombreuses requêtes SNMP alors que l'information peut être récupérée par une simple requête SNMP GET.

Pour envoyer cette requête, nous disposons de l'outil `snmpget` que nous pouvons utiliser comme précisé ci-après.

```
$ snmpget 192.168.1.1 .1.3.6.1.2.1.1.3.0
DISMAN-EVENT-MIB::sysUpTimeInstance = Timeticks: (220774) 0:36:47.74
```

On remarque que si on précise le numéro de l'objet sans y adjoindre l'identifiant '`0`', le client se voit répondre par l'agent SNMP que l'objet n'existe pas. En effet, tous les objets SNMP sont potentiellement des listes. Aussi, le fait d'adjoindre le chiffre 0 permet de récupérer la première entrée correspondant à cet OID.

Le résultat obtenu est une signature de temps avec une précision au centième de seconde près. Le client convertit automatiquement cette valeur en temps (ici, 36 minutes, 47 secondes et 74 centièmes).

5.2.4 Détermination des informations concernant l'interface Ethernet 0

Les informations concernant l'interface Ethernet 0 du routeur Cisco 2500 peuvent être obtenues à l'aide de `mbrowse`, par une navigation logique et graphique dans la table MIB ; ou à l'aide des commandes `snmpget` et `snmpgetnext` si on connaît les noms des objets concernés.

✖ Récupération à l'aide de mbrowse

Ne sachant pas où sont stockées les informations sur les interfaces du routeur Cisco dans la hiérarchie des objets SNMP, nous avons utilisé l'outil `mbrowse` pour trouver ces informations.

Tout d'abord, il est nécessaire d'identifier le type de chacune des interfaces réseaux du routeur. En effet, le routeur est équipé de plusieurs types d'interface et nous devons déterminer laquelle est la première. Pour cela, nous naviguons (WALK) dans les entrées de l'arborescence suivante, et obtenons le résultat ci-après :

`.iso.org.dod.internet.mgmt.mib-2.interfaces.ifTable.ifEntry.ifDescr.`

```
.iso.org.dod.internet.mgmt.mib-2.interfaces.ifTable.ifEntry.ifDescr.1 =
STRING: Ethernet0
.iso.org.dod.internet.mgmt.mib-2.interfaces.ifTable.ifEntry.ifDescr.2 =
STRING: Serial0
.iso.org.dod.internet.mgmt.mib-2.interfaces.ifTable.ifEntry.ifDescr.3 =
STRING: Serial1
.iso.org.dod.internet.mgmt.mib-2.interfaces.ifTable.ifEntry.ifDescr.4 =
STRING: TokenRing0
```

L'interface sur laquelle nous souhaitons obtenir des informations (`Ethernet 0`) est donc l'interface dont le numéro d'entrée est `1`. En observant cette même hiérarchie des interfaces réseaux (`interfaces`) de la machine, on observe qu'elle possède une clé `ifPhysAddress` qui stocke les adresses physiques (ici, MAC Ethernet) des interfaces.

Si on demande la première entrée de cette arborescence (entrée numéro `1`), on obtient le résultat ci-dessous :

```
.iso.org.dod.internet.mgmt.mib-2.interfaces.ifTable.ifEntry.ifPhysAddress.1
= STRING: 0:0:c:4a:d8:91
```

L'adresse MAC de l'interface `Ethernet0` est donc `00:00:0C:4A:D8:91`.

De même, on peut déterminer l'adresse IP de l'interface en utilisant l'objet `.iso.org.dod.internet.mgmt.mib-2.ip.ipAddrTable.ipAddrEntry.ipAdEntAddr`. L'utilisation d'un listage des sous entrées de cette entrée nous donne :

```
.iso.org.dod.internet.mgmt.mib-
2.ip.ipAddrTable.ipAddrEntry.ipAdEntAddr.192.168.1.1 = IpAddress:
192.168.1.1
.iso.org.dod.internet.mgmt.mib-
2.ip.ipAddrTable.ipAddrEntry.ipAdEntIfIndex.192.168.1.1 = INTEGER: 1
.iso.org.dod.internet.mgmt.mib-
2.ip.ipAddrTable.ipAddrEntry.ipAdEntNetMask.192.168.1.1 = IpAddress:
255.255.255.0
.iso.org.dod.internet.mgmt.mib-
2.ip.ipAddrTable.ipAddrEntry.ipAdEntBcastAddr.192.168.1.1 = INTEGER: 1
.iso.org.dod.internet.mgmt.mib-
2.ip.ipAddrTable.ipAddrEntry.ipAdEntReasmMaxSize.192.168.1.1 = INTEGER:
18024
```

On est assurée que l'adresse IP obtenue sur la première ligne (`192.168.1.1`) est bien celle de l'interface `Ethernet0` car l'entrée `ipAdEntIfIndex.192.168.1.1` retourne l'entier `1`, qui spécifie qu'il s'agit de la première interface listée dans l'objet `ifPhysAddress`. Ce numéro correspond bien à l'interface `Ethernet0`.

✖ Procédure à l'aide des commandes snmpget et snmpgetnext

Comme nous l'avons cité précédemment, la commande `snmpget` permet de récupérer une information particulière dans la hiérarchie SNMP, tandis que snmpgetnext permet de récupérer l'entrée suivante de la hiérarchie, à partir d'un identifiant d'objet connu.

Par exemple, pour obtenir la première entrée décrivant les interfaces réseaux, on utilise la commande suivante :

```
$ snmpgetnext 192.168.1.1 .iso.org.dod.internet.mgmt.mib-
2.interfaces.ifTable.ifEntry.ifDescr
IF-MIB::ifDescr.1 = STRING: Ethernet0
```

Nous observons ainsi que l'entrée `ifDescr.1` est donc bien l'interface `Ethernet0` recherchée. Avec `snmpget` on peut donc obtenir l'adresse MAC (matérielle) de la carte réseau `Ethernet0` :

```
$ snmpget 192.168.1.1 .iso.org.dod.internet.mgmt.mib-
2.interfaces.ifTable.ifEntry.ifPhysAddress.1
IF-MIB::ifPhysAddress.1 = STRING: 0:0:c:4a:d8:91
```

De même, pour obtenir l'adresse IP de cette interface, on utilise une requête SNMP GETNEXT formulée comme suit :

```
$ snmpgetnext 192.168.1.1 .iso.org.dod.internet.mgmt.mib-
2.ip.ipAddrTable.ipAddrEntry.ipAdEntAddr.0
IP-MIB::ipAdEntAddr.192.168.1.1 = IpAddress: 192.168.1.1
```

Cependant, le fait que la bonne adresse nous soit renvoyée est liée au fait que seule cette interface est configurée et ce cas de figure ne se présentera pas toujours. Pour vérifier que l'information correspond bien à l'interface dont nous souhaitons obtenir l'adresse IP, il faut envoyer une requête SNMP GET qui demande l'entrée `ipAdEntIfIndex` qui permette d'indiquer l'index de l'interface possédant une adresse IP donnée, en joignant à ce nom l'adresse IP pour laquelle on souhaite obtenir les informations, ici :

```
$ snmpget 192.168.1.1 .iso.org.dod.internet.mgmt.mib-
2.ip.ipAddrTable.ipAddrEntry.ipAdEntIfIndex.192.168.1.1
IP-MIB::ipAdEntIfIndex.192.168.1.1 = INTEGER: 1
```

Cette adresse IP est bien affectée à l'interface numéro `1`, c'est-à-dire, selon les informations ci-dessus, l'interface `Ethernet0`. Si cela ne correspondait pas, il aurait fallu faire un snmpgetnext sur l'objet `.iso.org.dod.internet.mgmt.mib-2.ip.ipAddrTable.ipAddrEntry.ipAdEntAddr.192.168.1.1` afin d'obtenir l'adresse IP suivante et vérifier à quelle interface elle a été affectée.

5.2.5 Codage des adresses IP en SNMP

Afin d'étudier la façon dont les adresses IP sont codées dans les paquets SNMP, nous devons étudier le contenu exact d'un paquet. Aucun outil Net-SNMP ne permet d'avoir ces informations car ils effectuent un décodage simplifiant la lecture des informations à l'utilisateur.

Pour faire cette manipulation, nous devons donc utiliser Ethereal/WireShark, qui nous permettra d'analyser les paquets réseaux échangés entre notre ordinateur et notre routeur. Nous obtenons la capture suivante :

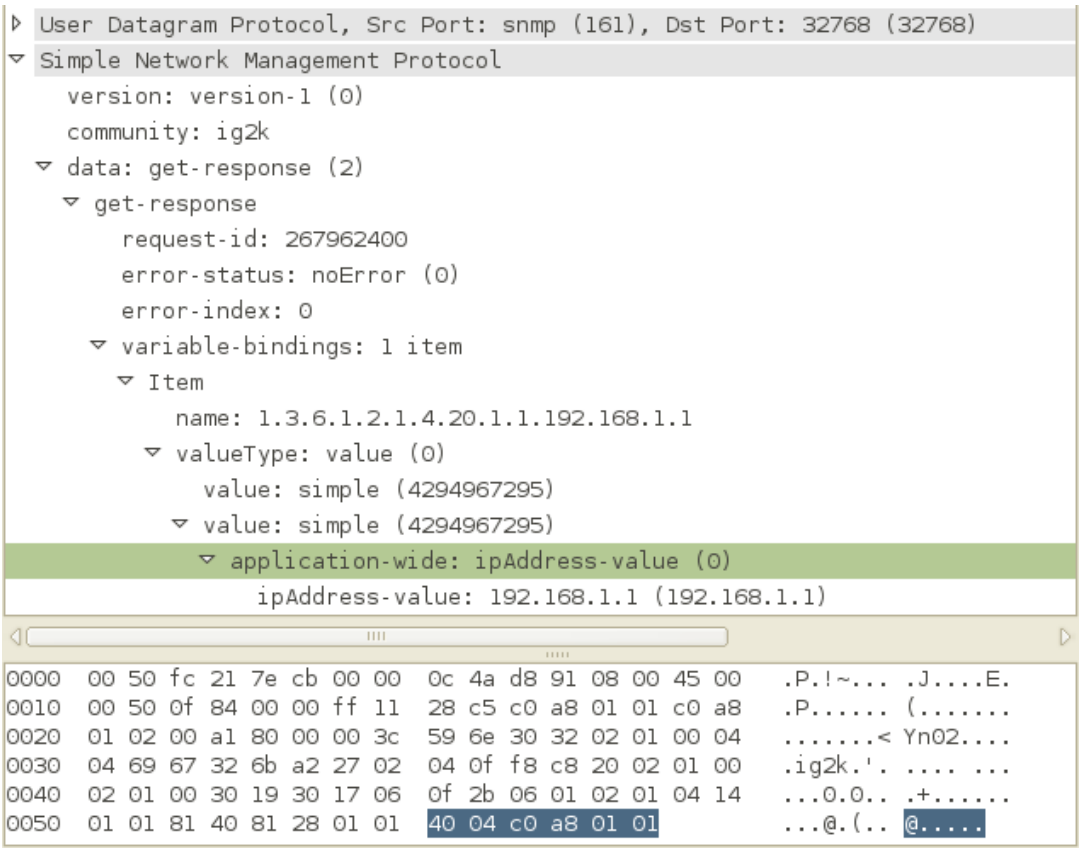

Figure 4 - Ethereal capture of ip address encoding SNMP GET packet

L'adresse IP est encodée dans le paquet en utilisant les conventions ASN-1. Le 1er octet désigne le numéro du type de l'objet renvoyé, qui ici est une adresse réseau (adresse IP en l'occurrence), de numéro `40`.

L'octet suivant désigne la longueur du contenu de ce type. Dans la mesure où nous sommes en IPv4, les adresses IP occupent 4 octets et cette valeur est donc `04`. Enfin, les 4 octets suivants désignent l'adresse IP, donnée en format binaire avec un découpage par octets qui sont organisés du MSB (octet le plus significatif) au LSB (octet le moins significatif). On appelle aussi cette organisation l'ordre des octets du réseau (*network byte order*).

En effet, $(c0)_{16} = (192)_{10}$, $(a8)_{16} = (168)_{10}$, $(01)_{16} = (1)_{10}$, ce qui correspond bien à l'adresse IP du routeur, obtenue par la consultation de l'objet SNMP `.1.3.6.1.2.1.4.20.1.1.192.168.1.1`.

Pour plus d'informations sur la notation ASN-1 et les conventions qui leur sont associées (qui est également utilisé pour le format de description des MIBs et SMIs), consultez le site du consortium ASN-1.

5.3 Contrôle distant à l'aide des outils SNMP

Comme nous l'avons vu précédemment, il est possible de configurer certains objets SNMP à l'aide de requêtes `SNMP SET`. Ces requêtes sont générées à l'aide de la commande `snmpset` et il est fréquent que le nom de la communauté ayant un accès en écriture ne soit pas le même que celle utilisée pour la lecture des informations.

Dans notre cas, les objets SNMP du routeur ne peuvent être modifiés que si on appartient à la communauté nommée `secret` et si on agit depuis l'adresse IP 192.168.1.2, comme nous l'avons configuré précédemment.

Nous souhaitons, afin de tester la modification de valeurs, désactiver l'interface `Ethernet0`, puis modifier la valeur de l'objet `sysLocation` du routeur (qui précise l'emplacement physique du routeur).

Le statut des interfaces du routeur (ou de la machine) étudié est stocké dans l'objet `ifAdminStatus`, qui est une collection qui stocke les informations sur le statut administratif de toutes les interfaces. La valeur de cet objet est un entier qui vaut 1 si l'interface est activée (`up`) et 2 si l'interface est désactivée (`down`).

Par exemple pour désactiver l'interface 1 (`Ethernet0`) on utilise la commande suivante :

```
$ snmpset -c secret 192.168.1.1 .iso.org.dod.internet.mgmt.mib-
2.interfaces.ifTable.ifEntry.ifAdminStatus.1 i 2

Timeout: No Response from 192.168.1.1
```

On observe que la modification a été immédiatement effective dans la mesure où le routeur ne répond plus car son interface avec laquelle nous communiquions avec lui est désormais désactivée.

Pour la réactiver, il est nécessaire d'utiliser notre connexion série par terminal `minicom` ou HyperTerminal. Si l'interface désactivée était une autre interface, nous pourrions la réactiver par l'envoi d'un message SNMP SET sur l'objet `ifAdminStatus` correspondant à l'interface et en demandant d'y placer l'entier (`i`) `1`.

Une fois l'interface `Ethernet0` du routeur réactivée, nous pouvons modifier les informations sur l'emplacement du routeur à l'aide de la commande `snmpset` comme suit :

```
$ snmpset -c secret 192.168.1.1 .iso.org.dod.internet.mgmt.mib-
2.system.sysLocation s "Nouvel emplacement du routeur"
```

La modification est effective instantanément et consultable via une requête GET : `snmpget 192.168.1.1 .iso.org.dod.internet.mgmt.mib-2.system.sysLocation`.

5.4 Gestion des alarmes (traps) SNMP

5.4.1 Vérification de la réception des alarmes SNMP

Le protocole SNMP permet aussi de définir des trap. Cela permet d'envoyer un paquet UDP à un serveur lorsqu'un événement donné se produit.

entête IP 20 o.	entête UDP 8 o.	version (0)	communauté	type PDU (4)	entreprise	adresse agent	type TRAP (0 à 5)	code spéc.	time stamping	T	L	V	…

Datagramme IP d'un TRAP SNMP

Source http://www.enseirb.fr/~kadionik/embedded/snmp/net-snmp.html

Il est possible de récupéré l'ensemble des alarmes émises par le routeur grâce à la commande snmptrapd. Sur cette capture, on remarque que l'événement 'Link Down Trap' est bien reçu lorsque l'interface 0 est réinitialisée.

```
# snmptrapd -Le -C -f
2007-02-27 15:43:15 NET-SNMP version 5.1.2 Started.


2007-02-27 15:44:23 192.168.1.1(via 192.168.1.1) TRAP, SNMP v1, community
ig2k
        SNMPv2-SMI::enterprises.9.1.29 Link Up Trap (0) Uptime: 1:29:45.66
        IF-MIB::ifIndex.1 = INTEGER: 1  IF-MIB::ifDescr.1 = STRING:
Ethernet0   IF-MIB::ifType.1 = INTEGER: ethernetCsmacd(6)   SNMPv2-
SMI::enterprises.9.2.2.1.1.20.1 = STRING: "up"
2007-02-27 15:45:01 192.168.1.1(via 192.168.1.1) TRAP, SNMP v1, community
ig2k
        SNMPv2-SMI::enterprises.9.1.29 Link Up Trap (0) Uptime: 1:30:23.97
        IF-MIB::ifIndex.1 = INTEGER: 1  IF-MIB::ifDescr.1 = STRING:
Ethernet0   IF-MIB::ifType.1 = INTEGER: ethernetCsmacd(6)   SNMPv2-
SMI::enterprises.9.2.2.1.1.20.1 = STRING: "up"
2007-02-27 15:45:52 192.168.1.1(via 192.168.1.1) TRAP, SNMP v1, community
ig2k
        SNMPv2-SMI::enterprises.9.9.43.2 Enterprise Specific Trap (1)
Uptime: 1:31:25.24
```

```
        SNMPv2-SMI::enterprises.9.9.43.1.1.6.1.3.6 = INTEGER: 1 SNMPv2-
SMI::enterprises.9.9.43.1.1.6.1.4.6 = INTEGER: 2 SNMPv2-
SMI::enterprises.9.9.43.1.1.6.1.5.6 = INTEGER: 3
2007-02-27 15:46:15 192.168.1.1(via 192.168.1.1) TRAP, SNMP v1, community
ig2k
        SNMPv2-SMI::enterprises.9.1.29 Link Down Trap (0) Uptime: 1:31:47.60
        IF-MIB::ifIndex.2 = INTEGER: 2  IF-MIB::ifDescr.2 = STRING: Serial0
IF-MIB::ifType.2 = INTEGER: propPointToPointSerial(22)  SNMPv2-
SMI::enterprises.9.2.2.1.1.20.2 = STRING: "administratively down"
```

5.4.2 Ajout de handlers invoqués lors de la réception d'alarmes

Pour exécuter une commande particulière pour une alarme donnée, il suffit d'éditer (ou de créer s'il n'existe pas) le fichier snmptrapd.conf.

Ce fichier doit respecter la syntaxe suivante :

```
traphandle            OID_du_trap         commande_à_exécuter argument
```

Ensuite, il faut définir le traphandle pour l'OID qui lancera un script. Pour ajouter des paramètres au script, le plus simple est de transmettre les variables sur l'entrée standard du script, par exemple :

```
traphandle            default             /tmp/cisco.sh
```

Il ne reste plus qu'a créer le script correspondant contenant l'ensemble des actions à effectuer lors de la réception de l'alarme.

5.4.3 Intérêt du système d'alarmes

Nous venons de voir qu'il est possible de configurer les traps pour attribuer un script à un signal. Ainsi dès que le signal sera reçu, le script sera automatiquement exécuté. L'utilisation des scripts permet une totale liberté d'action. On est ainsi capable d'envoyer des mails, lancer un programme de reboot automatique, envoyer des notifications de changement de route à d'autres machines…

5.5 Ajout des MIBs propriétaires Cisco

Cisco propose un certain nombre de Management Information Base et SMI spécifiques, qui sont propriétaires. Vous pouvez télécharger les définitions de ces MIB et SMI spécifiques aux adresses suivantes :

- http://monge.univ-mlv.fr/~roussel/SNMP/CISCO-CONFIG-MAN-MIB.my
- http://monge.univ-mlv.fr/~roussel/SNMP/CISCO-SMI.my
- http://monge.univ-mlv.fr/~roussel/SNMP/OLD-CISCO-SYS-MIB.my

Il est conseillé d'installer ces MIBs dans `/tmp/mibs` dans la mesure où nous ne les utiliseront que temporairement. Si vous souhaitez les utiliser en permanence, vous devriez les installer dans un répertoire persistant, car `/tmp` est automatiquement vidé à chaque démarrage de la machine.

5.5.1 Vérification de l'accès à l'arborescence Cisco

✖ Avant l'installation des MIBs propriétaires

Avant que les MIBs Cisco ne soient installés, la tentative de consultation de l'objet `.iso.org.dod.internet.private.enterprises.cisco.local.lsystem.romId.0` retournait l'erreur suivante :

```
Unknown Object Identifier (Sub-id not found: enterprises -> cisco)
```

Le système ne connaissant pas la hiérarchie cisco, il lui était impossible de convertir le nom de l'objet en un format numérique, qui puisse être transmis dans un paquet SNMP.

✖ Après l'installation des MIBs propriétaires

Une fois les MIBs installés comme décrit dans la section 0, il est possible de convertir les noms d'objets de l'architecture Cisco en numéros d'objets (OID numériques) et ainsi de parcourir ces informations spécifiques du routeur.

Ainsi, la requête précédente nous retourne désormais les informations demandées, ce qu'on peut vérifier tout d'abord en naviguant sur l'architecture propriétaire Cisco à l'aide de la commande `snmpwalk` :

```
$ snmpwalk 192.168.1.1 .iso.org.dod.internet.private.enterprises.cisco
```

5.5.2 Récupération de la version de l'IOS en utilisant l'arborescence privée

L'arborescence privée de Cisco permet d'obtenir de nombreuses informations propriétaires sur le routeur, le logiciel utilisé, les informations enregistrées etc. Afin de vérifier cela, nous allons récupérer comme information la version de la ROM utilisée au moment du boot, avant le chargement de l'IOS de Cisco.

Selon les informations fournies par Cisco, ces informations sont stockées dans l'objet `romId` récupérable comme suit :

```
$ snmpget 192.168.1.1
.iso.org.dod.internet.private.enterprises.cisco.local.lsystem.romId.0
OLD-CISCO-SYS-MIB::romId.0 = STRING: "..System Bootstrap, Version 5.2(5),
RELEASE SOFTWARE..Copyright (c) 1986-1994 by cisco Systems.."
```

L'identifiant numérique de l'objet peut être récupéré à l'aide de la commande `snmptranslate` comme suit :

```
$ snmptranslate -On
.iso.org.dod.internet.private.enterprises.cisco.local.lsystem.romId.0
.1.3.6.1.4.1.9.2.1.1.0
```

Conclusions

Ces travaux pratiques sur SNMP nous ont permis de découvrir ce protocole simple de surveillance et contrôle de périphériques réseaux à distance. Cette norme existant depuis plus de 20 ans, elle est implémentée sur de nombreux appareils réseaux et peut être installé sur la plupart des systèmes d'exploitations d'ordinateurs afin d'en permettre une gestion simplifiée.

Ainsi, il offre de nombreux avantages et de nombreuses possibilités comme la gestion semi-automatique d'un parc informatique, la configuration réseau à distance, l'envoi d'alertes en cas de problèmes sur un périphérique et ainsi une prise de décision manuelle (en avertissant un administrateur) ou automatique (en exécutant automatiquement une procédure de récupération sur le poste superviseur).

Bien entendu, les considérations concernant la sécurité et l'utilisation du protocole SNMP posent certains problèmes, dans la mesure où seules les dernières versions du protocole peuvent être considérées comme réellement sécurisée. Ainsi, se baser sur un nom de communauté partagé entre plusieurs machines et circulant en clair sur un réseau n'est pas une sécurité suffisante.

De plus, si on n'utilise pas de protections supplémentaires comme le filtrage des adresses IP autorisées à échanger des paquets SNMP, il est aisé pour un intrus d'envoyer des commandes distantes sur le réseau ou, plus simplement, de récupérer des informations sensibles sur la machine comme son emplacement physique ou son système d'exploitation et sa version.

Ces informations peuvent permettre d'effectuer soit des attaques physiques sur la machine, soit des attaques liées à la connaissance des failles d'une version donnée d'un système non maintenu. Aussi, le protocole SNMP doit il être utilisé avec parcimonie et être bien exploité pour être efficace, ce qui n'est pas du ressort de ce document.

Annexe A : Fichier de configuration snmpd.conf

```
##############################################################################
#
# This file is intended to only be an example.  If, however, you want
# to use it, it should be placed in /etc/snmp/snmpd.conf.
# When the snmpd agent starts up, this is where it will look for it.
#
# You might be interested in generating your own snmpd.conf file using
# the "snmpconf" program (perl script) instead.  It's a nice menu
# based interface to writing well commented configuration files.  Try it!


##############################################################################
# Access Control
##############################################################################

####
# First, map the community name (COMMUNITY) into a security name
# (local and mynetwork, depending on where the request is coming
# from):

#       sec.name  source          community
com2sec paranoid  default         public

####
# Second, map the security names into group names:

#               sec.model  sec.name
group MyROSystem v1        paranoid
group MyROSystem v2c       paranoid
group MyROSystem usm       paranoid
group MyROGroup v1         readonly
group MyROGroup v2c        readonly
group MyROGroup usm        readonly
group MyRWGroup v1         readwrite
group MyRWGroup v2c        readwrite
group MyRWGroup usm        readwrite

#           incl/excl subtree                          mask
view all    included  .1                               80
view system included  .iso.org.dod.internet.mgmt.mib-2.system

####
# Finally, grant the 2 groups access to the 1 view with different
# write permissions:

#                context sec.model sec.level match  read   write  notif
access MyROSystem ""      any       noauth    exact  system none   none
access MyROGroup ""       any       noauth    exact  all    none   none
access MyRWGroup ""       any       noauth    exact  all    all    none
```

```
# ----------------------------------------------------------------------------


##############################################################################
# System contact information
#

syslocation Unknown (configure /etc/snmp/snmpd.local.conf)
syscontact Root <root@localhost> (configure /etc/snmp/snmpd.local.conf)

# ----------------------------------------------------------------------------
```