Cours 1
Administration des réseaux : SNMP

Licence Pro: RIMS

Sylvain MERCHEZ

Plan

Administration des réseaux : SNMP

- Introduction
- Le protocole SNMP
- La MIB

Introduction

Simple Network Management Protocol permet de :

- visualiser une quantité d'informations concernant le matériel, les connexions réseaux, leur état de charge,
- modifier le paramétrage de certains composants,
- alerter l'administrateur en cas d'événements considérés comme grave, et d'autres choses encore...

Introduction : Le principe

Un agent sur chaque équipement va collecter les informations

petit service qui reste à l'écoute sur le port UDP 161

écoute et répond aux requêtes

modifie certaines informations

envoie des alertes

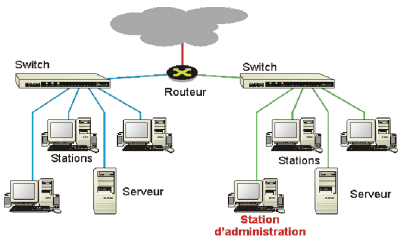
Introduction : Le principe

Un manager ou station d'administration

client SNMP (envoi de requêtes)

serveur à l'écoute des alertes (sur le port UDP 162)

Net-SNMP



Introduction : Le principe

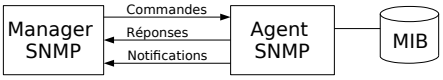
Exemple :

snmpwalk

```
# Quelques infos sur la machine et son OS :
SNMPv2-MIB::sysDescr.0 = STRING: Hardware: x86 Family 15 Model 2 Stepping 4
                         AT/AT COMPATIBLE - Software: Windows 2000 Version 5.1
                         (Build 2600 Uniprocessor Free)
...
# Le type de carte réseau :
IF-MIB::ifDescr.2 = STRING: D-Link DFE-528TX PCI Adapter
                    - Miniport d'ordonnancement de paquets
...
#Quelques infos sur la configuration IP :
IP-MIB::ipAdEntAddr.127.0.0.1 = IpAddress: 127.0.0.1
IP-MIB::ipAdEntAddr.192.168.0.10 = IpAddress: 192.168.0.10
IP-MIB::ipAdEntIfIndex.127.0.0.1 = INTEGER: 1
IP-MIB::ipAdEntIfIndex.192.168.0.10 = INTEGER: 2
IP-MIB::ipAdEntNetMask.127.0.0.1 = IpAddress: 255.0.0.0
IP-MIB::ipAdEntNetMask.192.168.0.10 = IpAddress: 255.255.255.0
...
RFC1213-MIB::ipRouteNextHop.0.0.0.0 = IpAddress: 192.168.0.250
...
#Les ressources de mémoire de masse :
HOST-RESOURCES-MIB::hrStorageDescr.2 = STRING: C:\ Label:  Serial Number d803c0ad
HOST-RESOURCES-MIB::hrStorageDescr.3 = STRING: D:\
HOST-RESOURCES-MIB::hrStorageDescr.4 = STRING: E:\ Label:DATA  Serial Number 10822ea3
HOST-RESOURCES-MIB::hrStorageDescr.5 = STRING: F:\
HOST-RESOURCES-MIB::hrStorageDescr.6 = STRING: G:\ Label:ZIP-100  Serial Number 1de00d03
...
# Les imprimantes installées :
HOST-RESOURCES-MIB::hrDeviceDescr.1 = STRING: HP DeskJet 600
HOST-RESOURCES-MIB::hrDeviceDescr.2 = STRING: HP DeskJet 550C
...
# Les services en cours d'exécution :
HOST-RESOURCES-MIB::hrSWRunName.1516 = STRING: "rundll32.exe"
HOST-RESOURCES-MIB::hrSWRunName.1520 = STRING: "OLFSUT40.EXE"
HOST-RESOURCES-MIB::hrSWRunName.1620 = STRING: "UATAPD32.EXE"
HOST-RESOURCES-MIB::hrSWRunName.1636 = STRING: "spampal.exe"
HOST-RESOURCES-MIB::hrSWRunName.2016 = STRING: "rundll32.exe"
HOST-RESOURCES-MIB::hrSWRunName.2125 = STRING: "msimn.exe"
...
# Les applications installées !!!
HOST-RESOURCES-MIB::hrSWInstalledName.9 = STRING: "Ethereal 0.9.8"
HOST-RESOURCES-MIB::hrSWInstalledName.10 = STRING: "Exodus Jabber Client (remove only)"
HOST-RESOURCES-MIB::hrSWInstalledName.11 = STRING: "FileZilla (remove only)"
...
HOST-RESOURCES-MIB::hrSWInstalledName.79 = STRING: "SpamPal: BadWords plugin"
HOST-RESOURCES-MIB::hrSWInstalledName.80 = STRING: "SpamPal"
...
# Ainsi que leur date d'installation...
HOST-RESOURCES-MIB::hrSWInstalledDate.9 = STRING: 2002-12-17,10:29:48.0
```

Le protocole SNMP

- Le protocole SNMP fait de la couche application dans le réseau Internet
- Il utilise le protocole UDP de la couche transport et
- Il est affecté aux ports 161 et 162 du protocole UDP par défaut



Le protocole : SNMP

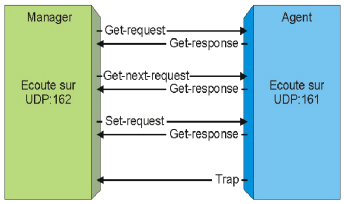
SNMP tire son "N" du fait qu'il s'appuie sur UDP d'une part, et qu'il ne propose qu'un nombre très restreint de commandes.

Les commandes sont les suivantes (version 1) :

Commande	Action
get-request	Le Manager SNMP demande une information à un agent SNMP
get-next-request	Le Manager SNMP demande l'information suivante à l'agent SNMP
set-request	Le Manager SNMP met à jour une information sur un agent SNMP
get-reponse	L'agent SNMP répond à un get-request ou a un set-request
trap	L'agent SNMP envoie une alarme au Manager

Agent Port UDP 161, Manager Port UDP 162

Le protocole : SNMP



Les commandes get-request, get-next-request et set-request sont toutes émises par le manager à destination d'un agent et attendent toutes une réponse get-response de la part de l'agent.

La commande trap est une alerte. Elle est toujours émise par l'agent à destination du manager, et n'attend pas de réponse.

Le protocole SNMP

Version	Community	SNMP PDU

message SNMP

PDU type	request id	0	0	variable bindings

GetRequest PDU, GetNextRequest PDU, SetRequest PDU

PDU type	request id	error status	error index	variable bindings

GetResponse PDU

PDU type	enterprise	agent addr	generic trap	specific trap	time stamp	variable bindings

Trap PDU

name1	Value1	name2	value2	···	namen	valuen

Variable bindings

SNMP : installation

Sous Linux : modifier le fichier /etc/snmp/snmpd.conf

Sous windows : dans Gestion de l'Ordinateur, Services et applications, Service SNMP (Onglet Sécurité)

Plan

Administration des réseaux : SNMP

- Introduction
- Le protocole SNMP
- La MIB

Organisation des informations : La MIB

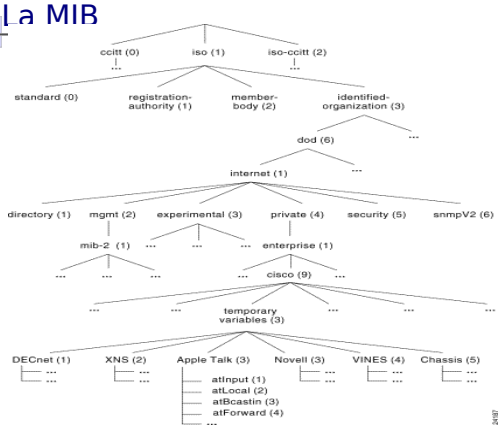
Qu'est ce ?

- Contient la base des informations de gestion
- Indépendante du matériel et du logiciel
- Base de données normalisée qui permet de lire et d'écrire sur les équipements distants.

Structure :

- organisée hiérarchiquement (sous forme d'arbre)
- contient une partie commune pour le même type de matériel et une partie spécifique pour le constructeur
- contient des scalaires ou des tableaux de scalaire
- appellation normalisée

Organisation des informations : La MIB



La MIB : un objet scalaire

- Un objet scalaire est un objet qui n'est pas contenu dans un tableau.
- Pour accéder à une instance(valeur concrète) d'un objet scalaire, le client ajoute '.0' à son identificateur (OID)

Exemple : la donnée qui représente le nombre d'erreurs pendant les connexions TCP est définie dans une MIB standard par l'OID suivant : .1.3.6.1.2.6.14

- Si un client (station d'administration) veut obtenir la valeur de l'instance de cette donnée, il faut indiquer dans le message la référence suivante : .1.3.6.1.2.6.14.0

La MIB : un objet scalaire

- Quelques exemples :

```
[root@gw mibs]# snmpget -v 1 -c public localhost .1.3.6.1.2.1.1.3.0
system.sysUpTime.0 = Timeticks: (23599841) 2 days, 17:33:18.41

[root@gw mibs]# snmpget -v 1 -c public localhost .1.3.6.1.2.1.1.3.1
Error in packet
Reason: (noSuchName) There is no such variable name in this MIB.
Failed object: system.sysUpTime.1

[root@gw mibs]# snmpget -v 1 -c public localhost 1.3.0
system.sysUpTime.0 = Timeticks: (23690312) 2 days, 17:48:23.12

[root@gw mibs]# snmpget -v 1 -c public localhost system.sysUpTime.0
system.sysUpTime.0 = Timeticks: (23721007) 2 days, 17:53:30.07
```

Outil : snmptranslate permet de faire la conversion entre le texte et l'index, nous informe sur le type et son état.

Un objet tableau

- Un objet tableau est un objet qui peut contenir plusieurs instances.

  SEQUENCE OF <entry>

  - Les objets (lignes d'une table ou *entry*) contenus dans le tableau peuvent être des séquences d'éléments (plusieurs éléments simples) .
  - Chaque séquence (ligne) du tableau doit contenir les mêmes éléments.

  SEQUENCE {<type1> ... <typeN>}

  - La taille de ce tableau n'est pas bornée.

Un objet tableau

- Pour accéder à une instance particulière du tableau, on ajoute à l'identificateur de l'objet une valeur d'index (qui peut être plus ou moins complexe).
  Par exemple l'objet
  ifDesc .1.3.6.1.2.1.2.2.1.2
  représente la deuxième colonne de la ligne
  ifEntry (.1.3.6.1.2.1.2.2.1)
  dans le tableau
  ifTable (.1.3.6.1.2.1.2.2),
  pour y accéder on utilisera l'identificateur
  .1.3.6.1.2.1.2.2.1.2.i
  avec i la valeur de l'index

Définition d'un objet d'une MIB

- Chaque objet dans une MIB suit la structure définie dans la syntaxe ASN.1

```
OBJECT-TYPE MACRO ::=
BEGIN
  TYPE NOTATION ::= "SYNTAX" type (TYPE ObjectSyntax)
                    "ACCESS" Access
                    "STATUS" Status
                    DescrPart
                    ReferPart
                    IndexPart
                    DefValPart
  VALUE NOTATION ::= value (VALUE ObjectName)

  Access ::= "read-only" | "read-write" | "write-only" | "not-accessible"
  Status ::= "mandatory" | "optional" | "obsolete" | "deprecated"
  DescrPart ::= "DESCRIPTION" value (description DisplayString) | empty
  DescrPart ::= "REFERENCE" value (reference DisplayString) | empty
  IndexPart ::= "INDEX" "{" IndexTypes "}"
  IndexTypes ::= IndexType | IndexTypes "," IndexType
  IndexType ::= value (indexobject ObjectName) | type(indextype)
  DefValPart ::= "DEFVAL" "{" value (defvalue ObjectSyntax) "}" | empty
  DisplayString ::= OCTET STRING SIZE (0..255)
END
```


Exemple d'un objet scalaire

- L'objet tcpMaxConn

```
tcpMaxConn OBJECT-TYPE
  SYNTAX INTEGER
  ACCESS read-only
  STATUS mandatory
  DESCRIPTION
      "The limit on the total number of TCP connections
       the entity can support. In entities where the
       maximum number of connections is dynamic, this
       object should contain the value -1."
  ::= {tcp 4}
```


Un exemple de table

- Le tableau tcpConnTable

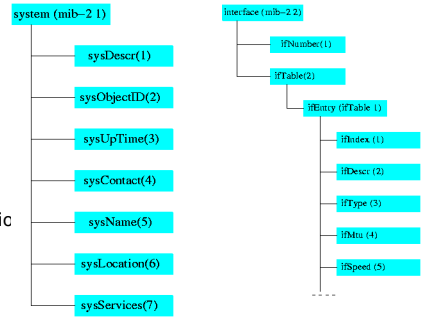
```
tcpConnTable OBJECT-TYPE
  SYNTAX SEQUENCE OF TcpConnEntry
  ACCESS not-accessible
  STATUS mandatory
  DESCRIPTION
    "A table containing TCP connection-specific information"
  ::= {tcp 13}
tcpConnEntry OBJECT-TYPE
  SYNTAX TcpConnEntry
  ACCESS not-accessible
  STATUS mandatory
  DESCRIPTION
    "..."
  INDEX {tcpConnLocalAddress, tcpConnLocalPort, tcpConnRemAddress, tcpConnRemPort}
  ::= {tcpConnTable 1}
TcpConnEntry ::= SEQUENCE { tcpConnState INTEGER,
                    tcpConnLocalAddress IPAddress,
                    tcpConnLocalPort INTEGER (0..65535)
                    tcpConnRemAddress IPAddress,
                    tcpConnRemPort INTEGER (0..65535)}
```


Les types des Objets d'une MIB

- Les types simples
  - INTEGER
  - OCTET STRING
  - OBJECT IDENTIFIER
  - NULL
  - SEQUENCE et SEQUENCE OF
- Les types applicatifs
  - NetworkAddress (CHOICE contenant uniquement IpAddress)
  - IpAdress
  - Counter (Counter32 ou Counter64 en SNMP2) repasse à 0 lorsque = max maximum (2^{32} - 1)
  - Gauge ne repasse pas à 0, maximum (2^{32} - 1)
  - TimeTicks compteur en 1/100 seconde depuis une origine
  - Opaque représente un codage arbitraire

La MIB standard (mib-2)

- La MIB mib-2 est le premier fils de mgmt dans l'arbre de nommage des objets. Cette MIB est constituée de 10 groupes:
  - System
  - Interfaces
  - At
  - Ip
  - Icmp
  - Tcp
  - Udp
  - Egp
  - Transmissic
  - Snmp



Les droits d'accès

- L'authentification
  - Dans chaque message échangé entre la station de supervision et les agents on trouve un nom de groupe d'utilisateurs (*community name*).
  - Le nom joue le rôle d'un mot de passe, c'est à dire que l'agent considère que si la station de supervision connaît le nom du groupe alors elle est authentifiée et autorisée à agir comme un membre du groupe.

Les droits d'accès

- La politique d'accès
  - L'agent peut limiter l'accès à une sélection de stations d'administration
  - L'agent peut définir plusieurs groupes (*community name*).
  - Le contrôle d'accès se compose de deux aspects
    - Une vue de la MIB: un sous ensemble de la MIB (plusieurs vues sont possibles pour chaque communauté)
    - Un mode d'accès {*read-only*, read-write}
    - La vue et le mode d'accès forment le profil de la communauté SNMP.

Les méthodes d'accès aux données gérées par un agent

- L'accès direct : exemple pour une table

TcpConnState (.1.3.6.1.2.1.6.13.1.1)	TcpConnLocalAddress (.1.3.6.1.2.1.6.13.1.2)	TcpConnLocalPort (.1.3.6.1.2.1.6.13.1.3)	TcpConnRemAddress (.1.3.6.1.2.1.6.13.1.4)	TcpConnRemPort (.1.3.6.1.2.1.6.13.1.5)	
5	10.0.0.99	12	9.1.2.3	15	TcpConnEntry(.1.3.6.1.2.1.6.13.1)
2	10.0.0.99	99	192.168.37.5	25	TcpConnEntry(.1.3.6.1.2.1.6.13.1)
3	10.0.0.99	14	89.1.1.42	84	TcpConnEntry(.1.3.6.1.2.1.6.13.1)

TcpConnState (.1.3.6.1.2.1.6.13.1.1)	TcpConnLocalAddress (.1.3.6.1.2.1.6.13.1.2)	TcpConnLocalPort (.1.3.6.1.2.1.6.13.1.3)	TcpConnRemAddress (.1.3.6.1.2.1.6.13.1.4)	TcpConnRemPort (.1.3.6.1.2.1.6.13.1.5)
x.1.10.0.0.99.12.9.1.2.3.15	x.2.10.0.0.99.12.9.1.2.3.15	x.3.10.0.0.99.12.9.1.2.3.15	x.4.10.0.0.99.12.9.1.2.3.15	x.5.10.0.0.99.12.9.1.2.3.15
x.1.10.0.0.99.14.89.1.1.42.84	x.2.10.0.0.99.14.89.1.1.42.84	x.3.10.0.0.99.14.89.1.1.42.84	x.4.10.0.0.99.14.89.1.1.42.84	x.5.10.0.0.99.14.89.1.1.42.84

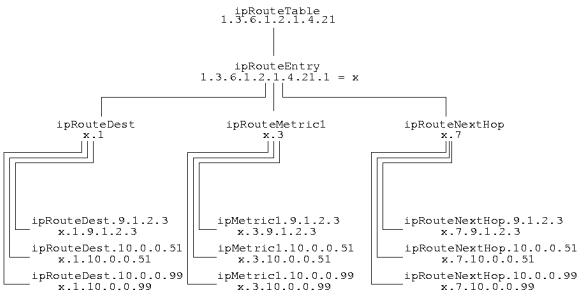
Les méthodes d'accès aux données gérées par un agent

- L'accès direct
  - Sélection de la colonne dans la table : on utilise l'identificateur (OID)
  - Sélection de la ligne dans la table : on utilise l'index défini pour la table
    - Soit y la valeur de l'OID de la donnée que l'on veut accéder
    - Soit i1, i2, ...iN les objets qui constituent l'index
    - Alors pour accéder à une colonne et une ligne particulière dans la table on utilisera l'identifiant suivant :
    - y.i1.i2.....iN

Les méthodes d'accès aux données gérées par un agent

- L'accès série
  - L'identificateur d'un objet porte dans son écriture (suite d'entiers) une représentation hiérarchique de la structure qui la contient.
  - La suite d'entiers permet d'utiliser un ordre lexicographique
  - Les noeuds fils sont définis en ajoutant un entier à la liste des entiers de l'identificateur du père et en visitant l'arbre de bas en haut et de la gauche vers la droite.
  - Cet accès série permet d'accéder à des objets sans en connaître l'identificateur exact.

Les méthodes d'accès aux données gérées par un agent



Echange sur le réseau pour le protocole SNMP

- Emission d'un message
  - Construction du PDU qui va contenir la requête
    - Type de message (get, get-next, set, réponse ou trap)
    - Génération d'un numéro d-identification de la requête
    - Liste des couples (variables ,valeur) que l'on veut échanger
  - Ajout d'un nom de communauté de la version
- Réception d'un message
  - Analyse du message
    - Examen du nom de la communauté
    - Examen du PDU (numéro de requête, les couples (variables, valeur)