Gestión de red en Linux con SNMP

Documento extraído de los artículos:
- "Gestión SNMP con Linux" por Javier Fernández-Sanguino Peña
- "Administración y mantenimiento de redes con Linux" por David Guerrero

1. Introducción a SNMP

En el mundo actual, en el que la informática gira en torno al concepto de red, el trabajo de los administradores de sistemas es muy complejo. Su misión consiste en mantener en funcionamiento los recursos así como cada dispositivo crítico que conforma la red.

Hay gran cantidad de motivos por los cuales un administrador necesita monitorizar entre otros: la utilización del ancho de banda, el estado de funcionamiento de los enlaces, la detección de cuellos de botella, detectar y solventar problemas con el cableado, administrar la información de encaminamiento entre máquinas, etc. La monitorización de la red es también un buen punto desde el que comenzar el estudio de los problemas de seguridad.

La respuesta a todas las necesidades antes expuestas, es el protocolo llamado SNMP (*Simple Network Management Protocol*). Su principal objetivo fue integrar la gestión de diferentes tipos de redes mediante un diseño sencillo y que produjera poca sobrecarga en la red.

SNMP opera en el nivel de aplicación, utilizando el protocolo de transporte TCP/IP, por lo que ignora los aspectos específicos del hardware sobre el que funciona. La gestión se lleva a cabo al nivel de IP, por lo que se pueden controlar dispositivos que estén conectados en cualquier red accesible desde la Internet, y no únicamente aquellos localizados en la propia red local. SNMP utiliza la capa de transporte de TCP/IP mediante el envío de datagramas UPD, sin embargo, el hecho de usar UDP hace que el protocolo no sea fiable (en UDP no se garantiza la recepción de los paquetes enviados, como en TCP).

El protocolo SNMP está compuesto por dos elementos: el agente y el gestor. Es una arquitectura cliente-servidor, en la cual el agente desempeña el papel de servidor y el gestor hace el de cliente.

El agente es un programa que ha de ejecutase en cada nodo de red que se desea gestionar o monitorizar. Ofrece un interfaz de todos los elementos que se pueden configurar. Estos elementos se almacenan en unas estructuras de datos llamadas MIB (*Management Information Base*), se explicarán más adelante. Representa la parte del servidor, en la medida que tiene la información que se desea gestionar y espera comandos por parte del cliente.

El gestor es el software que se ejecuta en la estación encargada de monitorizar la red, y su tarea consiste en consultar los diferentes agentes que se encuentran en los nodos de la red los datos que estos han ido obteniendo.

SNMP, también, permite a un agente enviar datos que no han sido solicitados de forma explícita al gestor, para informar de eventos tales como: errores, fallos en la alimentación eléctrica, etc.

La pregunta a responder es qué se puede hacer con SNMP. Con SNMP se puede monitorizar el estado de un enlace punto a punto para detectar cuando está congestionado y tomar así medidas oportunas, se puede hacer que una impresora alerte al administrador cuando se ha quedado sin papel, o que un servidor envíe una alerta cuando la carga de su sistema incrementa significativamente. SNMP también permite la modificación remota de la configuración de dispositivos, de forma que se podría modificar las direcciones IP de un ordenador a través de su agente SNMP, u obligar a la ejecución de comandos (si el agente ofrece las funcionalidades necesarias).

En esencia, el SNMP es un protocolo muy sencillo que consta de un juego de comandos reducido. Un gestor puede realizar sólo dos tipos diferentes de operaciones sobre un agente: leer o escribir un valor de una variable en el MIB del agente. Estas dos operaciones se conocen como petición-de-lectura (*get-request*) y petición-de-escritura (*set-request*). Hay un comando para responder a una petición-de-lectura llamado respuesta-de-lectura (*get-response*), que es utilizado únicamente por el agente.

La posibilidad de ampliación del protocolo está directamente relacionado con la capacidad del MIB de almacenar nuevos elementos. Si un fabricante quiere añadir un nuevo comando a un dispositivo, como puede ser un encaminador, tan sólo tiene que añadir las variables correspondientes a su base de datos (MIB).

Casi todos los fabricantes implementan versiones agente de SNMP en sus dispositivos: encaminadores, hubs, sistemas operativos, etc. Linux no es una excepción, existen varios agentes SNMP disponibles públicamente en la Internet.

1.1. Cuestiones de seguridad en SNMP

SNMP ofrece muy poco soporte para la autentificación. Tan sólo ofrece el esquema de dos palabras clave (*two-passwords*). La clave pública permite a los gestores realizar peticiones de valores de variables, mientras que la clave privada permite realizar peticiones de escritura. A estas palabras clave se les llama en SNMP *communities*. Cada dispositivo conectado con una red gestionada con SNMP, ha de tener configuradas estas dos *communities*.

Es muy común tener asignando por defecto el valor "public" al *community* público, y "private" al privado. Por lo que es muy importante cambiar estos valores para proteger la seguridad de tu red.

1.2. Base de datos MIB

SNMP define un estándar separado para los datos gestionados por el protocolo. Este estándar define los datos mantenidos por un dispositivo de red, así como las operaciones que están permitidas. Los datos están estructurados en forma de árbol; en el que sólo hay un camino desde la raíz hasta cada variable. Esta estructura en árbol se llama MIB (*Management Information Base*).

La versión actual de TCP/IP MIB es la 2 (MIB-II) y en ella se divide la información que un dispositivo debe mantener en ocho categorías (ver tabla 1). Cualquier variable ha de estar en una de estas categorías.

Categoría	Información
system	Información del host del sistema de encaminamiento
interfaces	Información de los interfaces de red
addr-translation	Información de traducción de direcciones
ip	Información sobre el protocolo IP
icmp	Información sobre el protocolo ICMP
tcp	Información sobre el protocolo TCP
udp	Información sobre el protocolo UDP
egp	Información sobre el protocolo (Exterior Gateway)

Tabla 1. Categorías TCP/IP

La definición de un elemento concreto MIB implica la especificación del tipo de dato que puede contener. Normalmente, los elementos de un MIB son enteros, pero también pueden almacenar cadenas de caracteres o estructuras más complejas como tablas. A los elementos de un MIB se les llama "objetos". Los objetos son los nodos hoja del árbol MIB, si bien, un objeto puede tener más de una instancia, como por ejemplo un objeto tabla. Para referirse al valor contenido en un objeto, se ha de añadir el número de la instancia. Cuando sólo exista una instancia del objeto, está es la instancia cero.

Por ejemplo, el objeto `ifNumber` de la categoría "interfaces" es un entero que representa el número de interfaces presentes en el dispositivo; mientras el objeto `ipRoutingTable` de la categoría "ip" contiene la tabla de encaminamiento del dispositivo.

Hay que acordarse de utilizar el número de la instancia para leer el valor de un objeto. En este caso, el número de interfaces presentes en un encaminador puede ser observado mediante la instancia `ifNumber.0`.

En el caso de ser un objeto tabla, se ha de utilizar el índice a la tabla como último número para especificar la instancia (fila de la tabla).

Existe otro estándar que define e identifica las variables MIB, llamado SMI (*Structure of Management Information*). SMI especifica las variables MIB, éstas se declaran empleando un lenguaje formal ISO llamado ASN.1, que hace que tanto la forma como los contenidos de estas variables sean no ambiguos.

El espacio de nombres ISO (árbol) está situado dentro de un espacio de nombres junto con otros árboles de otros estándares de otras organizaciones. Dentro del espacio de nombres ISO hay una rama específica para la información MIB. Dentro de esta rama MIB, los objetos están a su vez jerarquizados en subárboles para los distintos protocolos y aplicaciones, de forma que esta información puede representarse unívocamente.

La Figura 1 muestra el espacio de nombres del MIB del TCP/IP, éste está situado justo bajo el espacio del IAB "mgmt". La jerarquía también específica el número para cada nivel.

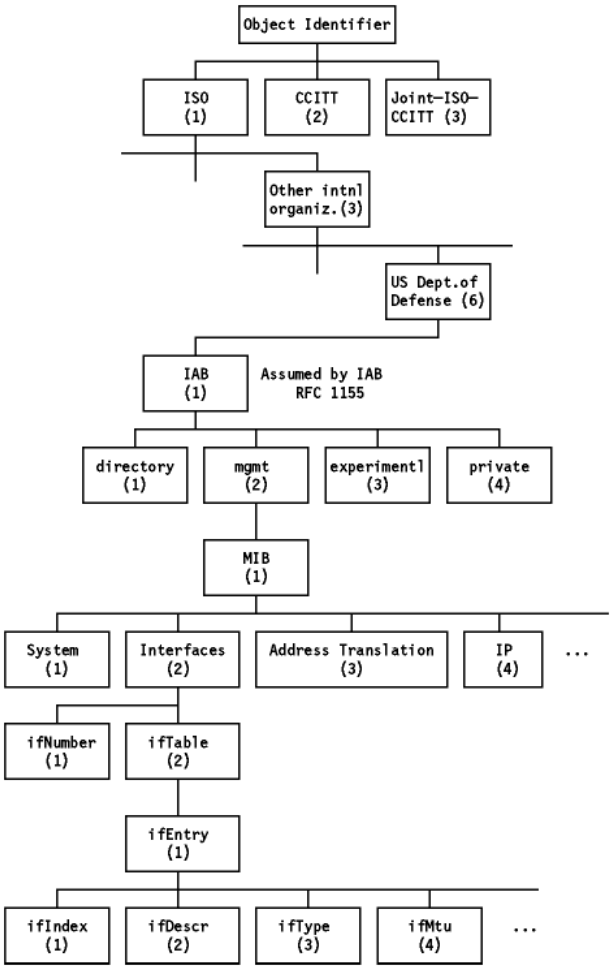

Figura 1. Espacio de nombres del MIB del TCP/IP

Es importante constatar que la mayor parte del software necesita el punto raíz (.) para localizar el objeto en el MIB. Si no se incluye el punto raíz, se asume que el path es relativo desde `.iso.org.dod.internet.mgmt.mib-2`.

De esta forma, el objeto `ifNumber` de la categoría "interfaces" se puede llamar:

```
.iso.org.dod.internet.mgmt.mib-2.interfaces.ifnumber
```

o el equivalente numérico:

```
.1.3.6.1.2.1.2.1
```

y la instancia es:

```
.iso.org.dod.internet.mgmt.mib-2.interfaces.ifnumber.0
```

o el equivalente numérico:

```
.1.3.6.1.2.1.2.1.0
```

Cada agente SNMP ofrece información dentro de una MIB, tanto de la general como de aquellas extensiones que desee proveer cada uno de los fabricantes. Así, los fabricantes de routers han extendido las MIBs estándar incluyendo información específica de sus equipos.

1.3. Cuestiones de versionado en SNMP

El SNMPv2 es un intento de solucionar las deficiencias que con el paso del tiempo han ido surgiendo en la primera versión. Estas deficiencias están causadas básicamente, por el intento de mantener la v1 lo más simple posible, además de por el incremento en la complejidad en las redes de ordenadores, y la necesidad de un protocolo más sofisticado. La aparición de esta segunda versión dio lugar a nuevos problemas, ya que al intentar mejorar deficiencias como, la falta de seguridad en las transmisiones, se llego a un protocolo demasiado complicado, por lo que ha sido revisado en varias ocasiones. Las características más destacables que se añaden en esta segunda versión son:

1. Seguridad: La primera versión no era segura, ya que se enviaba el nombre de la comunidad en texto llano, por lo que era posible hacer cambios en la configuración de un dispositivo por cualquier persona que tuviese acceso a la red. En la segunda versión se incorporan técnicas de encriptación para intentar solucionar este problema.
2. Operaciones con grandes volúmenes de información: La segunda versión permite a las estaciones administradoras la posibilidad de trabajar con grandes cantidades de datos, de una sola vez.
3. Nuevo formato para los eventos: Los traps en la primera versión tenían un formato diferente al de cualquiera de los otros comandos (`GET`, `SET` o `GETNEXT`). En la segunda versión se unifica este formato, siendo el mismo para un trap que para un comando de petición.
4. Comunicación administrador-administrador: Se extiende el protocolo para facilitar la comunicación entre diferentes estaciones administradoras, permitiendo la existencia de jerarquías que son necesarias en sistemas muy complejos. Este estilo de administración reduce el tráfico de administración y sobre todo la cantidad de información a transmitir.

2. Agentes SNMP en Linux

Ahora bien, es interesante saber qué puede hacer con SNMP y de qué herramientas dispone para hacerlo. Pues bien, con GNU/Linux y con herramientas de software libre se pueden hacer, entre otras cosas, las siguientes:

- instalar un agente SNMP para monitorizar variables en un servidor con GNU/Linux.
- utilizar en una estación con GNU/Linux una herramienta de gestión para observar variables de agentes SNMP.
- programar un interfaz para tomar medidas en base a la consulta (monitorización de variables de un elemento SNMP).
- programar un interfaz para recibir alertas SNMP y tratarlas como sea necesario.

En la mayoría de los sistemas GNU/Linux, se incluye un agente de SNMP que se trata de uno de los más desarrollados en la actualidad. Se trata de la actualización de la librería SNMP de la Universidad de California en Davis (que a su vez se basa en la librería de la Universidad de Carnegie Mellon). La librería se llamaba, en versiones previas, `ucd-snmp` pero ahora se denomina `net-snmp`. La versión actual ha sido portada a GNU/Linux de la librería original por Juergen Schoenwaelder y Erik Schoenfelder, el desarrollador principal es Wes Hardaker.

La versión actual incluye soporte para todas las versiones de SNMP (desde la uno, a la tres). Los agentes de SNMP que instala son perfectamente extensibles, tanto a través del propio código (con la API proporcionada) como a través de comandos definidos en la configuración.

Al tratarse de un software de agentes tan extendido, es conveniente detenerse un poco en su instalación y configuración, así como en las herramientas que proporciona.

2.1. Instalación de net-smp

Las distribuciones actuales, por ejemplo Debian ó Ubuntu, incorporan ya el paquete de `net-snmp` de forma que su instalación es mucho más sencilla (son binarios ya compilados) y su configuración rápida.

Lo primero que vamos a hacer es instalar el software:

```
$ apt-get install snmp snmpd
Desempaquetando snmp (de .../archives/snmp_4.2.3-2_i386.deb) ...
Seleccionando el paquete snmpd previamente no seleccionado.
Desempaquetando snmpd (de .../snmpd_4.2.3-2_i386.deb) ...
Configurando snmp (4.2.3-2) ...
Configurando snmpd (4.2.3-2) ...
Debian now uses the UCD SNMP agent/daemon. Since the new agent uses
an entirely new configuration file format, any configuration you may
have previously had can not be automatically updated and must be
replaced. Consequently, a security-conscious configuration will be
installed by default. Please read the snmpd.conf(5) manual page and
then edit /etc/snmp/snmpd.conf accordingly to change the configuration
to suit your needs.
Starting network management services: snmpd snmptrapd.
```

Lo anterior viene a decir que los paquetes se han instalado y configurado, y que por defecto me ha puesto una configuración segura.

De hecho, en la distribución se incluyen dos agentes. El primero `snmpd` es un agente que permanece escuchando en el puerto 161 (udp) esperando recibir peticiones, cuando le llega una solicitud la procesa y devuelve la información. El segundo, `snmptrapd` se trata de un agente que procesa las alertas de otros agentes. Para ello permanece escuchando en el puerto 162 (udp), cuando recibe una alerta por este puerto procede a guardarla en el registro (syslog). Sin embargo también puede ser configurado para utilizar programas externos en el tratamiento de las alertas.

Los agentes de net-snmp incluyen una serie de extensiones para poder obtener información específica del sistema como son:

- información general del sistema
- conexiones tcp/udp/ip/snmp abiertas y estado
- discos duros
- procesos y carga del procesador

2.2. Configuración de los agentes

Una vez instalados los agentes sólo será necesario adaptarlo a las necesidades del equipo en el que va a estar instalado. Por tanto, sólo queda poner una configuración adecuada en el fichero `/etc/snmp/snmpd.conf`.

Las primeras definiciones en el fichero de configuración definen las limitaciones para el acceso al agente desde cualquier servidor. El agente tiene soporte para la autenticación en SNMPv1, en SNMPv2c (con comunidades) y en SNMPv3 (a través de usuarios y grupos). Net-snmp implementa el Modelo de Control de Accesos Basados en Vistas (VACM, *View-Access Control Model*).

Lo primero que se debe definir es una relación entre comunidades y modelos de seguridad en el agente SNMP, tras esto se define una relación entre modelos de seguridad y grupos, se definen vistas (que son zonas del árbol de la MIB) y, finalmente, se indica el acceso permitido de los grupos a las vistas.

Esta tarea puede parecer compleja, por lo que quedará más claro con un ejemplo (fichero `/etc/snmp/snmpd.conf`) . Si se tiene definida la siguiente relación:

```
# sec.name source community
com2sec readonly default public
com2sec readwrite 127.0.0.1 private

# 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
```

Se está incluyendo todos los accesos como comunidad "public" desde cualquier lugar al grupo `MyROGroup`, mientras que los accesos como comunidad "private" desde el servidor local se vinculan al grupo `MyRWGroup`. Con las siguientes vistas definidas se termina la definición de los accesos a los agentes:

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

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

Con esta configuración garantizamos el acceso de escritura al grupo definido anteriormente (`MyRWGroup`) a cualquier parte de la MIB, mientras que sólo se permite leer dentro de la vista *system* (que está definida como una parte limitada de la MIB disponible) al grupo de sólo lectura.

Sin embargo a través de la configuración permite adaptar mucho más que sólo el acceso al agente. Entre otras cosas se puede:

- Hacer que el agente monitorice la existencia de procesos: `proc`. De esta forma se puede controlar que, por ejemplo, el proceso `apache` tenga más de 100 procesos. También es posible tomar acciones en caso de que las limitaciones impuestas a los procesos no se cumplan, definiéndolas con `procfix`.
- Hacer que el agente ejecute comandos con la función función `exec`. El agente ejecutará estos comandos cuando se consulte la variable de la MIB que se defina. De esta forma se puede utilizar el agente como una herramienta de gestión que tome acciones dentro del sistema, ampliando su comportamiento a algo más allá que un mero elemento que monitoriza variables en el sistema.
- Hacer que el agente controle la carga de la máquina para que se mantenga en unos límites determinados con el parámetro `load`.
- Definir algunos de los parámetros internos del agente en la MIB, como la ubicación del sistema (`syslocation`) o la persona de contacto (`syscontact`).
- Configurar el agente para enviar alertas a otros agentes cuando se den las condiciones necesarias. Para ello se tiene que definir la comunidad a utilizar con `trapcommunity` y el servidor concreto a utilizar con `trapsink`, `trap2sink`, ó `informsink`.

Como ya se ha comentado, el formato en detalle de la configuración de los agentes se puede consultar en el fichero de configuración `snmpd.conf`

2.3. Familiarizándose con el agente

Ya se debería tener el agente configurado y funcionando. Si no se ha lanzado aún habrá que lanzarlo ejecutando `/usr/sbin/snmpd`, la mayoría de las distribuciones instalarán un programa para poder parar y lanzar el demonio de forma sencilla. En el caso de Debian esto se consigue llamando el script `/etc/init.d/snmp` con la orden `start`.

Tras esto, llega el momento de familiarizarse con las herramientas de gestión SNMP incluidas dentro de `net-snmp`. Estas son:

- `snmpstatus` que permite acceder a la situación del agente.
- `snmpwalk` que permite *recorrer* la MIB del agente y sus variables.
- `snmpget` y `snmpset` que permiten, respectivamente, consultar y fijar atributos de SNMP.
- `snmptranslate` permite traducir de un identificador de objeto (OID) de la MIB a una cadena de caracteres representativa de éste.
- `snmpdelta`, establece un proceso de monitorización sobre una o más variables del agente, de forma que recuperar el valor de estas variables en periodos de tiempo definidos.
- `snmptest` es una herramienta de prueba del agente, al conectarse permite, a través de un interfaz de línea de comandos, recuperar cualquier variable que este contenga. Indica los métodos de comunicación usados contra el agente, por si fuera necesaria su depuración.
- `snmpnetstat`, es un comando atípico en las distribuciones de SNMP ya que es particular de la distribución `net-snmp`. Nos permite obtener un listado de los canales de comunicación abiertos en una máquina, al igual que `nestat`, pero utilizando un agente SNMP para recuperar la información.

Muchas de estas funciones son comunes de cualquier implementación de SNMP y el desarrollador las encontrará en cualquier distribución.

Así, si se desea saber si el agente está activo se haría:

```
$ snmpstatus -v 1 -c public localhost
[127.0.0.1]=>[Linux templar2.2.16-storm #1 Thu Aug 24 18:29:48 PDT 2000 i686] Up:
0:17:56.24
Interfaces: 0, Recv/Trans packets: 1908/1908 | IP: 1906/1906
```

Para consultar toda una rama se puede utilizar el comando `snmpwalk` un ejemplo de su uso se muestra en el listado 1. Para obtener un valor concreto del árbol (por ejemplo, la fecha del sistema) se ejecutaría:

```
$ snmpget -v 1 -c public localhost host.hrSystem.hrSystemDate.0
host.hrSystem.hrSystemDate.0 = 2001-2-12,18:51:20.0,+1:0
```

LISTADO 1 - Ejemplo de la salida del árbol con snmpwalk

```
system.sysDescr.0 = Linux templar 2.2.16-storm #1 Thu Aug 24 18:29:48 PDT 2000 i686
system.sysObjectID.0 = OID: enterprises.ucdavis.ucdSnmpAgent.linux
system.sysUpTime.0 = Timeticks: (121325) 0:20:13.25
system.sysContact.0 = Root >root@localhost<
system.sysName.0 = templar
system.sysLocation.0 = Mi casa
system.sysORLastChange.0 = Timeticks: (4) 0:00:00.04
system.sysORTable.sysOREntry.sysORID.1 = OID: ifMIB
system.sysORTable.sysOREntry.sysORID.2 = OID:
.iso.org.dod.internet.snmpV2.snmpModules.snmpMIB
system.sysORTable.sysOREntry.sysORID.3 = OID: tcpMIB
system.sysORTable.sysOREntry.sysORID.4 = OID: ip
system.sysORTable.sysOREntry.sysORID.5 = OID: udpMIB
system.sysORTable.sysOREntry.sysORID.6 = OID:
.iso.org.dod.internet.snmpV2.snmpModules.snmpVacmMIB.vacmMIBConformance.vacmMIBGroups.vac
mBasicGroup
system.sysORTable.sysOREntry.sysORID.7 = OID:
.iso.org.dod.internet.snmpV2.snmpModules.snmpFrameworkMIB.snmpFrameworkMIBConformance.snm
pFrameworkMIBCompliances.snmpFrameworkMIBCompliance
system.sysORTable.sysOREntry.sysORID.8 = OID:
.iso.org.dod.internet.snmpV2.snmpModules.snmpMPDMIB.snmpMPDMIBConformance.snmpMPDMIBCompl
iances.snmpMPDCompliance
system.sysORTable.sysOREntry.sysORID.9 = OID:
.iso.org.dod.internet.snmpV2.snmpModules.snmpUsmMIB.usmMIBConformance.usmMIBCompliances.u
smMIBCompliance
system.sysORTable.sysOREntry.sysORDescr.1 = The MIB module to describe generic objects
for network interface sub-layers
system.sysORTable.sysOREntry.sysORDescr.2 = The MIB module for SNMPv2 entities
system.sysORTable.sysOREntry.sysORDescr.3 = The MIB module for managing TCP
implementations
system.sysORTable.sysOREntry.sysORDescr.4 = The MIB module for managing IP and ICMP
implementations
system.sysORTable.sysOREntry.sysORDescr.5 = The MIB module for managing UDP
implementations
system.sysORTable.sysOREntry.sysORDescr.6 = View-based Access Control Model for SNMP.
system.sysORTable.sysOREntry.sysORDescr.7 = The SNMP Management Architecture MIB.
system.sysORTable.sysOREntry.sysORDescr.8 = The MIB for Message Processing and
Dispatching.
system.sysORTable.sysOREntry.sysORDescr.9 = The management information definitions for
the SNMP User-based Security Model.
system.sysORTable.sysOREntry.sysORUpTime.1 = Timeticks: (3) 0:00:00.03

(...)
```