

# Artículo para la revista Linux Actual número 17: "Gestión SNMP con Linux"

Javier Fernández-Sanguino Peña

12 Febrero 2001

---

En este artículo se verán a ver las distintas herramientas para utilizar un sistema GNU/Linux dentro de una red de gestión SNMP, tanto en la parte de agente como gestor.

---

## 1. Introducción a SNMP

SNMP (*Simple Network Management Protocol*) es el protocolo definido por los comités técnicos de Internet para ser utilizado como una herramienta de gestión de los distintos dispositivos en cualquier red. El funcionamiento de SNMP es sencillo, como dice el protocolo, aunque su implementación es tremendamente compleja. SNMP utiliza la capa de transporte de TCP/IP mediante el envío de datagramas UDP, 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 cubierto por un gran número de RFCs (*Request For Comments*), entre ellos el RFC 1157, 1215 (versión 1), del 1441 al 1452 (versión 2), del 2271 al 2275 y del 2570 al 2575 (para SNMP v3). El listado completo está disponible en <http://www.snmp.cs.utwente.nl/General/mngt-rfc.html>.

SNMP se basa en un conglomerado de agentes. Cada agente es un elemento de la red que ofrece unas determinadas variables al exterior, para ser leídas o modificadas. Asimismo, un agente puede enviar "alertas" a otros agentes para avisar de eventos que tengan lugar. Generalmente se llama "gestor" al agente encargado de recibir estos eventos.

El esquema es sencillo, sin embargo su complejidad se incrementa a la hora de definir las variables (y su formato). Las variables ofrecidas para consulta por los agentes SNMP se definen a través de una MIB (*Management Information Base*, Base de Información de Gestión). La MIB (hay sólo una aunque existen múltiples extensiones a ésta) es una forma de determinar la información que ofrece un dispositivo SNMP y la forma en que se representa. La MIB actual es MIB-II y está definida en el RFC 1213, aunque hay múltiples extensiones definidas en otros RFCs. La MIB está descrita en ASN.1 para facilitar su transporte transparente por la capa de red.

Cada agente SNMP ofrece información dentro de una MIB, tanto de la general (definida en los distintos RFCs) 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.

¿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)

## 2. SNMP en GNU/Linux

Ahora bien, SNMP es un mundo muy complejo y amplio y el lector posiblemente está interesado con saber qué puede hacer con SNMP en su servidor GNU/Linux 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.

Existen, también, herramientas propietarias para llevar a cabo estas funciones. Algunas de las más conocidas, como HP OpenView, SunNet Manager e IBM Netview, soportan muchas de las funciones que se van a tratar aquí. Sin embargo se va a entrar en detalle en herramientas de software libre, por considerarse que serán las herramientas más útiles para un desarrollador que quiera conocer "por dentro" el funcionamiento de los protocolos.

El estar en posesión del código fuente ayuda en gran medida a la persona que tiene que entrar en contacto con la tecnología ya que, rápidamente, puede familiarizarse con ésta a través de una implementación, o la puede poner a prueba compilándola e instalándola.

## 3. Agentes SNMP

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`, 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.

Esta librería ha sido muy actualizada y desarrollada e incluye las herramientas de SNMP "tradicionales". Las últimas versiones parten de la base de código de la versión 2.1 y han sido tremendamente mejoradas.

La versión actual, la 4.1, 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.

### 3.1 Instalación de net-smp

El primer paso será, sin duda, obtener el código fuente de la distribución. Anteriormente estaba disponible en <http://ucd-snmp.ucdavis.edu>, pero ahora se ha movido a sourceforge (para aprovechar los recursos que éste ofrece para proyectos libres) y está disponible en <http://www.sourceforge.net/projects/net-snmp>.

Una vez descargado y descomprimido en un directorio, se puede proceder a compilar el código fuente para ello se hace desde el raíz:

```
$ configure
$ make all
```

y, con suerte, quedará compilado y preparado para instalar. Esta librería no depende de otras, es autocontenida, lo que facilita su compilación. Lo cual se podrá hacer con `make install`. Ahora queda poner una configuración adecuada en el fichero `/etc/snmp/snmpd.conf`

Las distribuciones actuales, por ejemplo Debian ó RedHat, incorporan ya el paquete de `ucd-snmp` de forma que su instalación es mucho más sencilla (son binarios ya compilados) y su configuración rápida. Por ejemplo, para el paquete Debian de `ucd-snmp`, basta con instalarlo para tener ya un agente ejecutándose de forma transparente al

usuario.

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

## 3.2 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. La librería incluye una buena documentación que describe el formato de los ficheros de configuración.

En la página de manual `snmpd_config` se describe el funcionamiento general de los ficheros de configuración. En la instalación en sistemas Debian, el agente queda instalado con un fichero de ejemplo de configuración, en otros caso sera necesario copiar (o crear) uno en `/etc/snmp/snmpd.conf`. Pero muchos de los problemas pueden venir por no entender correctamente el modo de funcionamiento de la autenticación en SNMP.

Las primeras definiciones en el fichero de configuración definen las limitaciones para el acceso al agente desde cualquier servidor. Uno de los problemas más comunes es no ser capaz de acceder al agente por que estas restricciones son muy fuertes o no se han definido correctamente. El funcionamiento es, quizás, un tanto complejo, pero esto se debe a que 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*) definido como RFC.

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.

Esto puede parecer complejo, pero quedará más claro con un ejemplo. 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 `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`.

El formato en detalle de la configuración de los agentes se puede consultar en la página de manual `snmpd.conf`

## 3.3 Familiarizándose con el agente

Ya se debería tener el agente configurado y funcionando, hecho que se puede comprobar mirando el listado de procesos (con, por ejemplo, `ps aux |grep snmp`) y de conexiones para ver que hay un proceso `escuchando` en el puerto 161 (con, `netstat -anp -u`). 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.
- `snmpstat` 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 `netstat`, 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 localhost public
[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 localhost public host.hrSystem.hrSystemDate.0
host.hrSystem.hrSystemDate.0 = 2001-2-12,18:51:20.0,+1:0
```

### 3.4 Otros agentes SNMP para Linux

El agente de `net-snmp` no es el único agente disponible para los sistemas GNU/Linux aunque sí el que posiblemente se incluya en más distribuciones y esté más probado y extendido. Otros agentes a considerar dentro de GNU/Linux son:

- `snmpd-tcl` (disponible en <http://geekcorp.com/snmpd>). Se trata de una extensión mas que un agente en sí, para proveer la MIB de Recursos del Sistema (RFC 1514) dentro de un agente SNMP. Está pensado para poder multiplexar varios agentes a través del mismo puerto, de forma que cada uno ofrezca una MIB determinada.
- `opennms` el proyecto de Gestión Abierta de Redes (*Open Network Management*, [www.opennms.org](http://www.opennms.org)) ofrece una librería Java con licencia LGPL denominada JoeSNMP. Esta librería incluye una arquitectura completa de agentes SNMP desde el agente en sí a agentes para procesar las alarmas y un gestor genérico.
- `Agent++`, que es una implementación de agentes SNMP en C++ que soporta desde la versión 1 hasta la 3. Su licencia de distribución no es, sin embargo, libre (disponible en <http://www.agentpp.com>)
- `SNMP++`, al igual que el anterior está programado en C++, pero no soporta las mismas versiones de SNMP (de hecho es la base sobre la que se construyó `Agent++`). Su licencia de distribución tampoco es libre, disponible en <http://rosegarden.external.hp.com/snmp++/>

## 4. Herramientas para monitorizar agentes

Sin embargo un agente no sirve para mucho sin herramientas que lo monitoricen, esto lo dirá cualquier administrador. Dejando de un lado, temporalmente, el punto de vista del desarrollador y programador podemos pasar a responder la siguiente pregunta: ¿Qué necesita un administrador? Pues ni más ni menos que una herramienta visual, desde las que poder consultar las variables de los agentes, poner monitores para comprobar su evolución, y ver "gráficamente" las alertas.

Nuestro administrador de sistemas no tiene que pensar que no va a encontrar ésto aquí, y que ésto de GNU/Linux es sólo para los desarrolladores. Se le puede informar de que existen un buen número de herramientas disponibles para los sistemas GNU/Linux con soporte de SNMP (o que van a tenerlo pronto, según sus autores). Importante a destacar, y esto hará las delicias del personal de contabilidad, es que estas herramientas no tienen licencias de miles de euros (como sus equivalentes propietarias) y que no les van tan a la zaga en cuanto a características y funcionalidad disponible.

Entre otras herramientas podemos hablar de :

- `Scotty` (también conocido como `tkined`), es una herramienta completa de monitorización incluye capacidades de gestión/monitorización de dispositivos SNMP. Está implementada en Tcl/Tk, con extensiones propias, e incluye hasta un navegador de MIBs. Disponible en <http://wwwsnmp.cs.utwente.nl/~schoenw/scotty/>
- `Softguard`. Se trata de un navegador para agentes SNMP y sus MIBs, incluye funciones de auto descubrimiento y está también implementado en Tcl/Tk aunque está mucho más orientado hacia SNMP que `Scotty`. Se puede obtener de <http://www.osn.de/user/finzel/html/sgSpies.html>
- `NetraMet`. Se trata de una herramienta diseñada para gestionar el accounting de servidores (RFC 1272, 2063, 2064 y 2123). Incluye una implementación para tratar las extensiones de NetFlow de CISCO. Descargable desde <http://www.auckland.ac.nz/net/NeTraMet/>
- `Gxsnmp` es un gestor de elementos SNMP aún en desarrollo pero con un gran potencial. Las versiones actuales son betas muy recientes que carecen de un gran número de funcionalidades. Sin embargo sus bases son sólidas. Utiliza una base de datos para almacenar la información de agentes y redes (tiene interfaces programados a varias, entre otras, mysql), y la librería SMI para acceder a las MIBs. Es posible que, en un futuro y cuando sea más madura, sea la aplicación por excelencia para gestión de agentes SNMP en entornos GNU/Linux. Se puede obtener en <http://www.gxsnmp.org>
- `Gkrellm` es un monitor que permite monitorizar múltiples características del sistema, desde la capacidad del disco al uso de la CPU con un bonito widget en el escritorio. Existe una extensión a este monitor que permite incorporar variables SNMP para monitorizarlas junto con el resto de características del sistema.
- `mrtg`. Herramienta con interfaz WWW que permite una lectura en tiempo real de estadísticas de distintos elementos, entre otros, dispositivos SNMP. Es una de las herramientas más conocidas para monitorización de tráfico, y una de las más extendidas. Consultar [www.mrtg.org](http://www.mrtg.org)
- `cheops`. Herramienta sustitutiva de `scotty` para la gestión de elementos de red, aún no incluye soporte de SNMP pero es tremendamente gráfica e intuitiva.
- `mon`. Se trata de una herramienta integrada para la gestión de red, soportando múltiples sistemas en los que, a través de agentes, se pueden monitorizar las aplicaciones de éstos y su rendimiento. Tiene soporte de SNMP y ofrece la posibilidad de definir muchos niveles de alertas, desde correo electrónico a notificaciones con voz en tiempo real. Disponible en <http://www.kernel.org/software/mon>.
- `big brother`. Pretende ser una herramienta integrada, con una interfaz orientada a WWW para monitorizar estadísticas. Su licencia no permite la distribución comercial, pero aún así tiene características interesantes, aunque se ve superada por las anteriores.

## 5. Desarrollo de agentes

No se puede terminar este artículo sin estudiar la forma de integrar la gestión SNMP con aplicaciones a través de la programación de interfaces de acceso SNMP. De no hacerlo así es probable que algún programador experimentado se sienta defraudado por no haber llegado hasta el último detalle.

Pues bien, aunque existan herramientas, como ya se han visto, de gestión de dispositivos, un desarrollador puede querer generar su propia aplicación a medida para acceder a valores ofrecidos por elementos SNMP. De hecho esto puede ser incluso un requisito de un cliente y quizás la solución no este disponible entre los elementos que ya hemos destacado.

Actualmente se pueden desarrollar aplicaciones con acceso a SNMP utilizando librerías libres en tres lenguajes distintos: Perl, Tcl/Tk, PHP, C y Java.

En el caso de Perl existen tres librerías distintas de acceso a SNMP, la librería `Net::SNMP` (disponible en [cpan.perl.org](http://cpan.perl.org)), la librería `SNMP_session` (disponible en <ftp://ftp.switch.ch/software/sources/network/snmp/perl/>) y el módulo de extensión de Perl para la librería UCD SNMPv3 (disponible en <ftp://ftp-east.baynetworks.com/netman/snmp/perl5>). Salvo la última, estas librerías no necesitan tener ninguna librería de agentes instalada en el sistema.

Para Tcl existe la extensión proporcionada por el interfaz de gestión `scotty` denominada `Tnm::snmp` (`Tnm` es la extensión para herramientas de gestión de red) que permite incorporar funciones de gestión de SNMP tanto dentro de `scotty` como fuera de éste. También el lenguaje PHP (versión 3 y versión 4) incorpora extensiones con módulos SNMP para poder programa aplicaciones con interfaces WWW con acceso a dispositivos de red.

Para las versiones compiladas (C y Java) se puede utilizar cualquiera de las librerías de desarrollo mencionadas anteriormente (`net-snmp` u `openmms`) ya que cualquiera de ellas ofrece una API completa para el acceso a las funciones de SNMP.

## 5.1 Ejemplo de una aplicación en Perl

Por último se van a ver algunos ejemplos de aplicaciones utilizando la librería `Net::SNMP`. Se utiliza ésta librería por estar mejor orientada a objetos que las demás librerías en Perl, y por estar bien documentada. Nótese que la versión de la librería de Perl integrada con la librería de `net-snmp` tiene soporte para las últimas versiones de SNMP (v3). Se ha elegido también la primera de éstas por no depender de la implementación concreta de SNMP en el sistema.

En el listado 2 se puede ver un ejemplo de aplicación con Perl. Esta aplicación tan sólo recoge una variable SNMP del agente ejecutándose en el sistema local, dicha variable viene descrita por su identificador de objeto (los números punteados de la MIB).

Como se puede ver, lo primero que se hace es crear un objeto sesión, entregando los parámetros necesarios para generar la sesión (servidor al que se va a contactar, comunidad y puerto). Posteriormente, se llama a la función de consulta de variables `get_request`. Tras la comprobación de errores correspondiente se muestra el resultado y se cierra la sesión.

La librería puede operar de dos modos: bloqueante o no-bloqueante. En modo bloqueante las peticiones se ejecutan por el orden indicado y no se continúa el flujo del programa hasta que se lleva a cabo. En el modo no-bloqueante se puede indicar la función encargada de procesar los datos y continuar la ejecución del programa. Dado que las peticiones tardan tiempo en ser recibidas por el agente SNMP y devueltas, puede pensarse en situaciones de comprobaciones de múltiples agentes (o múltiples variables en múltiples agentes) en las que lo mejor sea enviar todas las solicitudes a los agentes y procesar las respuestas en paralelo, en lugar de ir una a una. Éste tipo de programación es la que se puede realizar con el modo bloqueante.

El resto de las funciones que ofrece el objeto sesión son similares variando, quizás, algunos parámetros. Una función también interesante es `trap` que permite enviar alertas a otros agente SNMP. Con esta función se puede implementar fácilmente, como se muestra en el Listado 3.

Para entrar en detalles en la programación de accesos a SNMP es necesario, casi obligatorio, entrar en detalles del funcionamiento y acceso a las MIBs de los agentes. Este tema queda, sin embargo, fuera del ámbito de este artículo.

## 6. Resumen

Se ha dado una visión general de qué es es SNMP y cómo puede ser utilizado éste, tanto desde el punto de vista de un agente incluido en el sistema operativo para monitorizar sus acciones a el desarrollo de aplicaciones con acceso a SNMP pasando someramente por algunas de las herramientas de monitorización disponibles.

**7. Sumarios SNMP es un protocolo de gestión SNMP se basa en conjuntos de agentes Con GNU/Linux se pueden tener y monitorizar agentes SNMP Net-snmp es una de las librerías de agentes más notables Existen herramientas propietarias pero son menos útiles a un desarrollador. Los agentes net-snmp son perfectamente extensibles. La definición de accesos puede resultar compleja. Es conveniente familiarizarse con las herramientas de SNMP. Hay más agentes disponibles para GNU/Linux. Existen herramientas para que el administrador monitorice los agentes. Gxsnmp es un gestor en desarrollo con un gran potencial El desarrollo de aplicaciones que utilizen SNMP es sencillo. Existen librerías para acceso a SNMP en Perl, Tcl/Tk, PHP, C y Java La librería libnet-snmp-perl está bien orientada a objetos y es sencilla de usar. Para entrar en detalles de programación es necesario conocer las MIBs**

## 8. Listados

### LISTADO 1-

```
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.vacmMIBGroupsConformance
system.sysORTable.sysOREntry.sysORID.7 = OID: .iso.org.dod.internet.snmpV2.snmpModules.snmpFrameworkMIB.snmpFrameworkMIBConformance.snmpFrameworkMIBConformance
system.sysORTable.sysOREntry.sysORID.8 = OID: .iso.org.dod.internet.snmpV2.snmpModules.snmpMPDMIB.snmpMPDMIBConformance.snmpMPDMIBCompliance
system.sysORTable.sysOREntry.sysORID.9 = OID: .iso.org.dod.internet.snmpV2.snmpModules.snmpUsmMIB.usmMIBConformance.usmMIBCompliance
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
(...)
```

PIE LISTADO 1: Ejemplo de la salida del árbol con `snmpwalk`

### LISTADO 2-

```
use strict;
use vars qw($session $error $response);

use Net::SNMP;

($session, $error) = Net::SNMP->session(
    -hostname => shift || 'localhost',
    -community =>: shift || 'public',
    -port => shift || 161
```

```

);

if (!defined($session)) {
    printf("ERROR: %s.\n", $error);
    exit 1;
}

my $sysUpTime = '1.3.6.1.2.1.1.3.0';

if (!defined($response = $session->get_request($sysUpTime))) {
    printf("ERROR: %s.\n", $session->error());
    $session->close();
    exit 1;
}

printf("sysUpTime para el servidor '%s' es %s\n",
    $session->hostname(),
    $response->{$sysUpTime}
);

$session->close();
exit 0;

```

PIE LISTADO 2: Ejemplo de captura de información SNMP con Perl

LISTADO 3-

```

use strict;
use vars qw($session $error $response);

use Net::SNMP;

($session, $error) = Net::SNMP->session(
    -hostname => shift || 'localhost',
    -community => shift || 'public',
    -port     => shift || 162
);

if (!defined($session)) {
    printf("ERROR: %s.\n", $error);
    exit 1;
}

my $count = 0;

$response = 1;
while ( $response ) {
    $response = $session->trap();
    $count++;
    sleep 5;
    print "Sending trap ($count)\n";
}

$session->close();
exit 0;

```

PIE LISTADO 3: Ejemplo de envío de alertas con Perl

LISTADO 4

Estos son algunos enlaces útiles generales para SNMP:

- [snmpwalk.org](http://www.snmpwalk.org/src/Tools.html). Sobre todo destaca la relación de herramientas para SNMP en <http://www.snmpwalk.org/src/Tools.html>
- [snmpworld.com](http://www.snmpworld.com)
- La FAQ de SNMP, disponible de <ftp://rtfm.mit.edu/pub/usenet/comp.protocols.snmp/>
- El servidor del proyecto de Gestión Abierta de Redes: [www.opennms.org](http://www.opennms.org)
- El HOWTO de Networking en Linux, disponible en [www.linuxdoc.org/LDP/](http://www.linuxdoc.org/LDP/)
- Un artículo, un tanto desactualizado, pero interesante, firmado por David Guerrero sobre SNMP y Linux, disponible en <http://www.david-guerrero.com/papers/snmp/>
- Un listado (no actualizado, pero útil) de herramientas de administración para Linux: <http://linas.org/linux/NMS.html>
- El grupo de noticias: [comp.protocols.snmp](mailto:comp.protocols.snmp)

Por supuesto hay multitud de libros en relación con SNMP, consulte la FAQ para ver una buena relación de éstos.

PIE LISTADO 4: Más información

## 9. Capturas

- Captura 1. [gxsnp-browser.jpg](#). El navegador gráfico de MIBs de gxsnp.
- Captura 2. [gxsnp-netmap2.jpg](#). Gestión de elementos de red con gxsnp.
- Captura 3. [mrtg.gif](#). Ejemplo de estadísticas de MRTG.
- Captura 4. [sgimospy.png](#). Acceso a un agente SNMP desde Softguard.
- Captura 5. [sgmibspy.png](#). Acceso a las MIBs desde Softguard.
- Captura 6. [gkrellm\\_snmp.jpg](#). Monitor integrado con SNMP gkrellm

## 10. Notas de maquetación

Las capturas son ejemplo de las aplicaciones de monitorización listadas en el epígrafe "Herramientas para monitorizar agentes" convendría que se situaran en el mismo lugar.

## 11. Notas de coordinación