

Trabajo Práctico

TPL 3 - Simple Network Management Protocol (SNMP)

Fecha de entrega: 13/10/2021

Franco, Juan Martín 149.615

juanmartin_franco@hotmail.com

Experiencia de laboratorio

1. ¿Qué podría hacer para descubrir cuales equipos de una red determinada tienen el servicio SNMP disponible? Busque agentes en la red de su hogar ¿Encontró algún equipo?

Para descubrir equipos que tengan el servicio SNMP disponible podría realizar, por ejemplo, un escaneo con nmap de tipo UDP (es decir, con el parámetro -sU) y buscar aquellos equipos que tengan el puerto 161 y/o 162 abiertos (los correspondientes a SNMP y SNMP para recepción de traps).

2. Instale en su equipo el paquete snmp para obtener los clientes que permitirán interactuar contra agentes remotos mediante tal protocolo.

Para instalar el paquete snmp, es necesario ejecutar el siguiente comando:

```
sudo apt-get install snmp
```

```
juanma@ubuntu:~$ sudo apt-get install snmp
[sudo] password for juanma:
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
  snmp
0 upgraded, 1 newly installed, 0 to remove and 54 not upgraded.
Need to get 168 kB of archives.
After this operation, 685 kB of additional disk space will be used.
Get:1 http://us.archive.ubuntu.com/ubuntu focal-updates/main amd64 snmp amd64 5.8+dfsg-2ubuntu2.3 [168 kB]
Fetched 168 kB in 1s (150 kB/s)
Selecting previously unselected package snmp.
(Reading database ... 183911 files and directories currently installed.)
Preparing to unpack .../snmp_5.8+dfsg-2ubuntu2.3_amd64.deb ...
Unpacking snmp (5.8+dfsg-2ubuntu2.3) ...
Setting up snmp (5.8+dfsg-2ubuntu2.3) ...
Processing triggers for man-db (2.9.1-1) ...
```

Comandos SNMP - Sintaxis y ejemplo de uso

`snmpget -v2c -c COMUNIDAD AGENTE OID`

`snmpwalk -v2c -c COMUNIDAD AGENTE [OID]`

`snmpset -v2c -c COMUNIDAD AGENTE OID TIPO VALOR`

Donde:

COMUNIDAD es el nombre de la comunidad con la que se identificará el cliente

AGENTE es la dirección IP o nombre de host a quien se consultará

OID es el objeto a consultar (o en blanco para solicitar todo)

TIPO es el tipo de datos del objeto (s=string, i=integer, ...)

VALOR es el valor que se desea asignar al objeto

-m ALL hace que el comando resuelva el OID a nombre, de ser posible

```
$ snmpget -v2c -c public localhost iso.3.6.1.2.1.1.0
```

```
iso.3.6.1.2.1.1.0 = STRING: "Linux geopistol 3.13.0-32-generic #57-Ubuntu SMP"
```

```
$ snmpwalk -v2c -c public -m ALL 170.210.101.102 SNMPv2-MIB::sysName
```

```
SNMPv2-MIB::sysName.0 = STRING: 409-Samsung
```

3. Por suerte, o por una mala decisión de administración de red, existen en internet agentes snmp que responden consultas a la comunidad publica. Para esta práctica vamos a aprovecharlos. Para comenzar, realice consultas SNMP al dispositivo cuya dirección IP ha obtenido en la clase para determinar:

a. ¿Qué dispositivo es? ¿Qué marca y modelo? ¿Qué OID u OIDs tuvo que consultar para obtener tal información (indique el número completo y la denominación en texto)?

Para obtener el tipo de dispositivo, utilicé el siguiente comando:

```
snmpget -v2c -c public 198.12.32.85 1.3.6.1.2.1.1.5.0
```

```
juanma@ubuntu:~$ snmpget -v2c -c public 198.12.32.85 1.3.6.1.2.1.1.5.0
iso.3.6.1.2.1.1.5.0 = STRING: "MikroTik"
```

Para obtener la marca y el modelo, utilicé el siguiente comando:

```
snmpget -v2c -c public 198.12.32.85 1.3.6.1.2.1.1.1.0
```

```
juanma@ubuntu:~$ snmpget -v2c -c public 198.12.32.85 1.3.6.1.2.1.1.1.0
iso.3.6.1.2.1.1.1.0 = STRING: "RouterOS RB3011UiAS"
```

Toda esta información puede ser visualizada mediante un único comando, el cual es:

```
snmpwalk -v2c -c public 198.12.32.85 1.3.6.1.2.1.1
```

```
juanma@ubuntu:~$ snmpwalk -v2c -c public 198.12.32.85 1.3.6.1.2.1.1
iso.3.6.1.2.1.1.1.0 = STRING: "RouterOS RB3011UiAS"
iso.3.6.1.2.1.1.2.0 = OID: iso.3.6.1.4.1.14988.1
iso.3.6.1.2.1.1.3.0 = Timeticks: (13673500) 1 day, 13:58:55.00
iso.3.6.1.2.1.1.4.0 = ""
iso.3.6.1.2.1.1.5.0 = STRING: "MikroTik"
iso.3.6.1.2.1.1.6.0 = ""
iso.3.6.1.2.1.1.7.0 = INTEGER: 78
```

Resumiendo, es un router cuyo fabricante es Mikrotik y su modelo es RB3011UiAS.

La denominación en texto es la siguiente:

iso(1).org(3).dod(6).internet(1).mgmt(2).mib-2(1).system(1)

b. ¿Qué otra información, que considera útil, podría ser recuperada del agente?

Otra información que podría resultar útil para recuperar:

- Las tablas de ruteo:

```
snmpwalk -v2c -c public 198.12.32.85 1.3.6.1.2.1.4.21
```

```
juanma@ubuntu:~$ snmpwalk -v2c -c public 198.12.32.85 1.3.6.1.2.1.4.21
iso.3.6.1.2.1.4.21.1.1.0.0.0.0 = IPAddress: 0.0.0.0
iso.3.6.1.2.1.4.21.1.1.5.5.5.0 = IPAddress: 5.5.5.0
iso.3.6.1.2.1.4.21.1.1.10.0.17.0 = IPAddress: 10.0.17.0
iso.3.6.1.2.1.4.21.1.1.10.0.18.0 = IPAddress: 10.0.18.0
iso.3.6.1.2.1.4.21.1.1.10.0.41.0 = IPAddress: 10.0.41.0
iso.3.6.1.2.1.4.21.1.1.10.0.46.0 = IPAddress: 10.0.46.0
iso.3.6.1.2.1.4.21.1.1.10.3.3.0 = IPAddress: 10.3.3.0
iso.3.6.1.2.1.4.21.1.1.10.6.6.0 = IPAddress: 10.6.6.0
iso.3.6.1.2.1.4.21.1.1.10.7.7.0 = IPAddress: 10.7.7.0
iso.3.6.1.2.1.4.21.1.1.10.10.2.0 = IPAddress: 10.10.2.0
iso.3.6.1.2.1.4.21.1.1.10.10.4.0 = IPAddress: 10.10.4.0
```

- La cantidad de interfaces del dispositivo:

```
snmpwalk -v2c -c public 198.12.32.85 1.3.6.1.2.1.2.1
```

```
juanma@ubuntu:~$ snmpwalk -v2c -c public 198.12.32.85 1.3.6.1.2.1.2.1
iso.3.6.1.2.1.2.1.0 = INTEGER: 14
```

- El valor por defecto del TTL de la cabecera IP de los datagramas originados en el host:

```
snmpwalk -v2c -c public 198.12.32.85 1.3.6.1.2.1.4.2
```

```
juanma@ubuntu:~$ snmpwalk -v2c -c public 198.12.32.85 1.3.6.1.2.1.4.2
iso.3.6.1.2.1.4.2.0 = INTEGER: 255
```

c. ¿Qué procedimientos y herramientas utilizó para descubrir los OID que resultan interesantes?

Para descubrir los OID que me resultaron interesantes utilicé 2 métodos:

1. Ir escalando el árbol hacia los nodos padres, o sea, borrando las hojas (siempre utilizando el comando snmpwalk).
2. Buscando en la página de [OIDView](#) (si conozco la marca del dispositivo que quiero monitorear) o en [OIDRef](#) (si quiero recorrer el árbol seleccionando la rama).

4. Instalar el paquete **snmp-mibs-downloader** y descargar las bases mediante la herramienta **download-mibs**. Editar el archivo **/etc/snmp/snmp.conf** y comentar la línea existente con **#**.

El archivo debería contener entonces solo la línea **# mibs** :

Para realizar esto, utilizo el comando → `sudo apt-get install snmp-mibs-downloader`

```

juanma@ubuntu:~$ sudo apt-get install snmp-mibs-downloader
[sudo] password for juanma:
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  smstrip
The following NEW packages will be installed:
  smstrip snmp-mibs-downloader
0 upgraded, 2 newly installed, 0 to remove and 54 not upgraded.
Need to get 5,170 kB of archives.
After this operation, 5,410 kB of additional disk space will be used.
Do you want to continue? [Y/n] Y
Get:1 http://us.archive.ubuntu.com/ubuntu focal/universe amd64 smstrip all 0.4.8+dfsg2-16 [7,904 B]
Get:2 http://us.archive.ubuntu.com/ubuntu focal/multiverse amd64 snmp-mibs-downloader all 1.2 [5,162 kB]

```

Ahora, procedo a editar el archivo `/etc/snmp/snmp.conf` con el comando:

```
sudo nano /etc/snmp/snmp.conf
```

```

GNU nano 4.8 /etc/snmp/snmp.conf
# As the snmp packages come without MIB files due to license reasons, loading
# of MIBs is disabled by default. If you added the MIBs you can reenale
# loading them by commenting out the following line.
# mibs :

# If you want to globally change where snmp libraries, commands and daemons
# look for MIBS, change the line below. Note you can set this for individual
# tools with the -M option or MIBDIRS environment variable.
#
# mibdirs /usr/share/snmp/mibs:/usr/share/snmp/mibs/iana:/usr/share/snmp/mibs/ietf

```

Repetir las consultas anteriores. ¿Qué diferencia aprecia? ¿A qué se debe?

Al repetir las consultas anteriores, obtengo los siguientes resultados:

```

juanma@ubuntu:~$ snmpwalk -v2c -c public 198.12.32.85 1.3.6.1.2.1.4.1
IP-MIB::ipForwarding.0 = INTEGER: forwarding(1)
juanma@ubuntu:~$ snmpwalk -v2c -c public 198.12.32.85 1.3.6.1.2.1.4.2
IP-MIB::ipDefaultTTL.0 = INTEGER: 255
juanma@ubuntu:~$ snmpwalk -v2c -c public 198.12.32.85 1.3.6.1.2.1.5.2
IP-MIB::icmpInErrors = No Such Object available on this agent at this OID
juanma@ubuntu:~$ snmpwalk -v2c -c public 198.12.32.85 1.3.6.1.2.1.2.1
IF-MIB::ifNumber.0 = INTEGER: 14

```

La diferencia que se aprecia es que, una vez realizada la consulta, la respuesta que se recibe contiene el nombre del OID que estoy solicitando.

Es decir, en lugar de recibir:

iso.3.6.1.2.1.4.2,

recibo:

IP-MIB::ipDefaultTTL.0, lo cual es mucho más amigable para el usuario.

Esto se debe a que el paquete que descargamos previo a la resolución de este ejercicio contiene información acerca del árbol de las MIBs y sus ramas.

5. Utilizar los comandos `snmpwalk` y `snmpbulkwalk` para consultar la rama `system` del mismo dispositivo. Realizar una captura para cada una de las ejecuciones, medir el tiempo de ejecución de ambos comandos y contrastarlos. Recuerde que el comando `time`, antepuesto a otro, permite realizar ésta medición (`man time` para más información).

Guardar las capturas con el nombre **snmpwalk.pcapng** y **snmpbulkwalk.pcapng** respectivamente y anotar las diferencias de tiempo obtenidas.

Para realizar esto, utilicé los siguientes comandos:

```
time snmpwalk -v2c -c public 198.12.32.85 1.3.6.1.2.1.1
```

```
time snmpbulkwalk -v2c -c public 198.12.32.85 1.3.6.1.2.1.1
```

Obteniendo el siguiente resultado:

Snmpwalk:

1	0.000000000	192.168.66.131	198.12.32.85	SNMP	83	get-next-request	1.3.6.1.2.1.1
2	0.042640823	198.12.32.85	192.168.66.131	SNMP	104	get-response	1.3.6.1.2.1.1.0
3	0.042916359	192.168.66.131	198.12.32.85	SNMP	85	get-next-request	1.3.6.1.2.1.1.1.0
4	0.095617415	198.12.32.85	192.168.66.131	SNMP	93	get-response	1.3.6.1.2.1.1.2.0
5	0.096058068	192.168.66.131	198.12.32.85	SNMP	85	get-next-request	1.3.6.1.2.1.1.2.0
6	0.165161435	198.12.32.85	192.168.66.131	SNMP	89	get-response	1.3.6.1.2.1.1.3.0
7	0.165930054	192.168.66.131	198.12.32.85	SNMP	85	get-next-request	1.3.6.1.2.1.1.3.0
8	0.200001224	198.12.32.85	192.168.66.131	SNMP	85	get-response	1.3.6.1.2.1.1.4.0
9	0.201069836	192.168.66.131	198.12.32.85	SNMP	85	get-next-request	1.3.6.1.2.1.1.4.0

Snmpbulkwalk:

5	12.300652499	192.168.66.131	198.12.32.85	SNMP	83	getBulkRequest	1.3.6.1.2.1.1
6	12.340424932	198.12.32.85	192.168.66.131	SNMP	261	get-response	1.3.6.1.2.1.1.0 1.3.6.1.2.1.1.2.0 1.3.6.1.2.1.1.3.0 1.3.6.1.2.1.1.4.0

```
variable-bindings: 10 items
  1.3.6.1.2.1.1.1.0: 526f757465724f532052423330313155694153
  1.3.6.1.2.1.1.2.0: 1.3.6.1.4.1.14988.1 (iso.3.6.1.4.1.14988.1)
  1.3.6.1.2.1.1.3.0: 28834400
  1.3.6.1.2.1.1.4.0: <MISSING>
  1.3.6.1.2.1.1.5.0: 4d696b726f54696b
  1.3.6.1.2.1.1.6.0: <MISSING>
  1.3.6.1.2.1.1.7.0: 78
  1.3.6.1.2.1.2.1.0: 14
  1.3.6.1.2.1.2.2.1.1.1: 1
  1.3.6.1.2.1.2.2.1.1.2: 2
```

Como se puede apreciar, el comando **snmpbulkwalk** se ejecuta mucho más rápido que el comando **snmpwalk**, esto se debe a que el comando **snmpbulkwalk** trae todos los resultados juntos, en cambio el **snmpwalk** busca un resultado, lo devuelve y luego busca el siguiente, haciendo 10 búsquedas separadas (por ejemplo, suponiendo que la rama tiene 10 hojas).

6. Buscar en Internet la MIB que corresponde a información de sistemas de alimentación ininterrumpible (UPS-MIB). Utilizando dicha MIB, consultar a la UPS cuya IP es

a. Información del voltaje de entrada.

```
juanma@ubuntu:~$ snmpwalk -v2c -c public 201.148.106.3 1.3.6.1.4.1.935.1.1.1.3.2.1
SNMPv2-SMI::enterprises.935.1.1.1.3.2.1.0 = INTEGER: 2190
```

b. Carga remanente en las baterías.

```
juanma@ubuntu:~$ snmpwalk -v2c -c public 201.148.106.3 1.3.6.1.4.1.935.1.1.1.2.2.4
SNMPv2-SMI::enterprises.935.1.1.1.2.2.4.0 = INTEGER: 0
```

c. Minutos estimados que durará la UPS si se interrumpe la energía.

```
juanma@ubuntu:~$ snmpwalk -v2c -c public 201.148.106.3 1.3.6.1.4.1.935.1.1.1.2.1.2
SNMPv2-SMI::enterprises.935.1.1.1.2.1.2.0 = INTEGER: 0
```

d. Indique los OID que utilizó y los valores obtenidos en cada caso.

Los OID utilizados fueron los siguientes:

Información del Voltaje de entrada: 1.3.6.1.4.1.935.1.1.1.3.2.1

Carga remanente en las baterías: 1.3.6.1.4.1.935.1.1.1.2.2.4

Minutos estimados que durará la UPS si se interrumpe la energía: 1.3.6.1.4.1.935.1.1.1.2.1.2

7. Buscar en la documentación el OID que corresponde a la tabla de interfaces (puertos) de un dispositivo de red. Utilizando el comando `snmptable` y dicho OID; obtener el listado de los puertos del switch `190.228.30.253` y su estado. Guardar la salida en un archivo denominado `tablintinterfaces-switch.txt` y analizar las columnas obtenidas.

El OID correspondiente a la tabla de interfaces es el siguiente: 1.3.6.1.2.1.2.2

Dicho OID se corresponde con: iso.identified-organization.dod.internet.mgmt.mib-2.interface.ifTable

Para obtener el listado de puertos del switch y su estado, utilizo el comando:

```
snmptable -v2c -c public 190.228.30.253 1.3.6.1.2.1.2.2
```

Obteniendo el siguiente archivo:

SNMP table: IF-MIB::ifTable

ifIndex	ifDescr	ifType	ifMtu	ifSpeed	ifPhysAddress	ifAdminStatus	ifOperStatus	ifLastChange	ifInOctets	ifInUcastPkts	ifInNUcastPkts	ifInDiscards
1	port1	iso88023Csmacd	1500	1000000000	00:00:cd:24:5a:9b	up	up	0:0:00:04.74	1283784341	1470722452	232778870	0
2	port2	iso88023Csmacd	1500	0	00:00:cd:24:5a:9b	up	down	0:0:00:00.00	0	0	0	0
3	port3	iso88023Csmacd	1500	1000000000	00:00:cd:24:5a:9b	up	up	244:2:24:59.04	1271983851	3843425455	19740	0
4	port4	iso88023Csmacd	1500	0	00:00:cd:24:5a:9b	up	down	0:0:00:00.00	0	0	0	0
5	port5	iso88023Csmacd	1500	0	00:00:cd:24:5a:9b	up	down	0:0:00:00.00	0	0	0	0
6	port6	iso88023Csmacd	1500	0	00:00:cd:24:5a:9b	up	down	0:0:00:00.00	0	0	0	0
7	port7	iso88023Csmacd	1500	1000000000	00:00:cd:24:5a:9b	up	up	165:4:46:26.72	633721985	2959731241	1296726	0
8	port8	iso88023Csmacd	1500	0	00:00:cd:24:5a:9b	up	down	0:0:00:00.00	0	0	0	0
9	port9	iso88023Csmacd	1500	0	00:00:cd:24:5a:9b	up	down	0:0:00:00.00	0	0	0	0
10	port10	iso88023Csmacd	1500	0	00:00:cd:24:5a:9b	up	down	0:0:00:00.00	0	0	0	0
11	port11	iso88023Csmacd	1500	0	00:00:cd:24:5a:9b	up	down	0:0:00:00.00	0	0	0	0
12	port12	iso88023Csmacd	1500	0	00:00:cd:24:5a:9b	up	down	0:0:00:00.00	0	0	0	0
13	port13	iso88023Csmacd	1500	0	00:00:cd:24:5a:9b	up	down	0:0:00:00.00	0	0	0	0
14	port14	iso88023Csmacd	1500	0	00:00:cd:24:5a:9b	up	down	0:0:00:00.00	0	0	0	0
15	port15	iso88023Csmacd	1500	0	00:00:cd:24:5a:9b	up	down	0:0:00:00.00	0	0	0	0
16	port16	iso88023Csmacd	1500	0	00:00:cd:24:5a:9b	up	down	0:0:00:00.00	0	0	0	0
17	port17	iso88023Csmacd	1500	0	00:00:cd:24:5a:9b	up	down	0:0:00:00.00	0	0	0	0
18	port18	iso88023Csmacd	1500	0	00:00:cd:24:5a:9b	up	down	0:0:00:00.00	0	0	0	0
19	port19	iso88023Csmacd	1500	0	00:00:cd:24:5a:9b	up	down	0:0:00:00.00	0	0	0	0
20	port20	iso88023Csmacd	1500	0	00:00:cd:24:5a:9b	up	down	0:0:00:00.00	0	0	0	0
21	port21	iso88023Csmacd	1500	0	00:00:cd:24:5a:9b	up	down	0:0:00:00.00	0	0	0	0
22	port22	iso88023Csmacd	1500	0	00:00:cd:24:5a:9b	up	down	0:0:00:00.00	0	0	0	0
23	port23	iso88023Csmacd	1500	0	00:00:cd:24:5a:9b	up	down	0:0:00:00.00	0	0	0	0
24	port24	iso88023Csmacd	1500	1000000000	00:00:cd:24:5a:9b	up	up	0:0:00:03.42	3903417500	3086682003	27218232	0
25	vlan1	12vlan	1500	0	00:00:cd:24:5a:9b	up	up	0:0:00:03.34	3652560796	36831286	230733030	3594578
26	conectividad	ieee8023adLag	1500	0	00:00:cd:24:5a:9b	up	down	0:0:00:00.06	0	0	0	0

De la tabla obtenida se pueden sacar las siguientes conclusiones:

1. El switch tiene 26 puertos.
2. Los 26 puertos están habilitados para operar (columna ifAdminStatus).
3. Del total de puertos, solo 5 forman un enlace (columna ifOperStatus).
4. El switch posee una Vlan la cual está asignada en el puerto 25.
5. Es un switch Ethernet (el tamaño de la MTU según la columna ifMtu es 1500, el correspondiente a Ethernet).

8. Para continuar con la experiencia de laboratorio y pasar a realizar acciones que modifiquen la configuración de los equipos se va a utilizar un laboratorio virtual diseñado para tal fin. Para ello descargue, descomprima e inicie el laboratorio de Netkit SNMPv2 desde el enlace siguiente

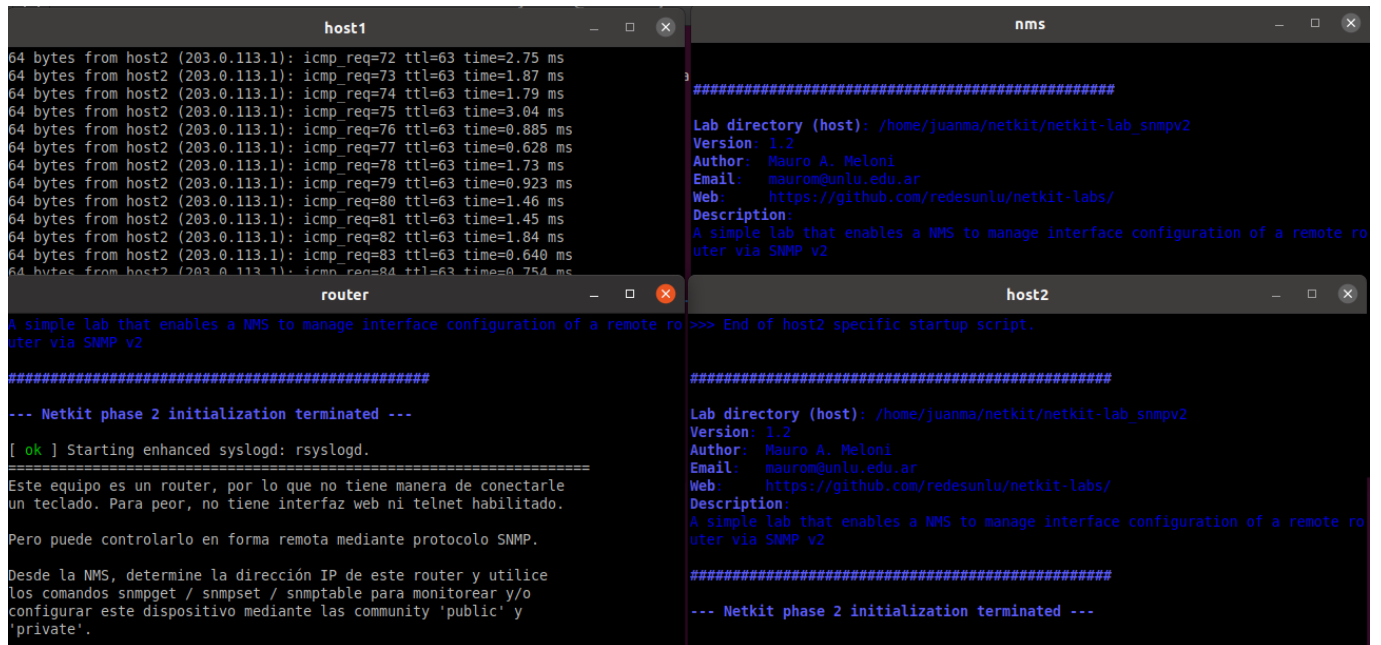
https://github.com/redesunlu/netkit-labs/raw/master/tarballs/netkit-lab_snmpv2.tar.gz.

Si encuentra dificultades en la utilización de Netkit o de este laboratorio en particular, repase las guías disponibles en <https://github.com/redesunlu/netkit-doc> o bien contacte al equipo docente.

Una vez descargado el laboratorio desde el link provisto, procedo a iniciarlo situándome en la carpeta del mismo (desde la terminal) y utilizando el comando:

```
lstart
```

Obteniendo lo siguiente:



```
host1
64 bytes from host2 (203.0.113.1): icmp_req=72 ttl=63 time=2.75 ms
64 bytes from host2 (203.0.113.1): icmp_req=73 ttl=63 time=1.87 ms
64 bytes from host2 (203.0.113.1): icmp_req=74 ttl=63 time=1.79 ms
64 bytes from host2 (203.0.113.1): icmp_req=75 ttl=63 time=3.04 ms
64 bytes from host2 (203.0.113.1): icmp_req=76 ttl=63 time=0.885 ms
64 bytes from host2 (203.0.113.1): icmp_req=77 ttl=63 time=0.628 ms
64 bytes from host2 (203.0.113.1): icmp_req=78 ttl=63 time=1.73 ms
64 bytes from host2 (203.0.113.1): icmp_req=79 ttl=63 time=0.923 ms
64 bytes from host2 (203.0.113.1): icmp_req=80 ttl=63 time=1.46 ms
64 bytes from host2 (203.0.113.1): icmp_req=81 ttl=63 time=1.45 ms
64 bytes from host2 (203.0.113.1): icmp_req=82 ttl=63 time=1.84 ms
64 bytes from host2 (203.0.113.1): icmp_req=83 ttl=63 time=0.640 ms
64 bytes from host2 (203.0.113.1): icmp_req=84 ttl=63 time=0.754 ms

router
A simple lab that enables a NMS to manage interface configuration of a remote router via SNMP v2

--- Netkit phase 2 initialization terminated ---

[ ok ] Starting enhanced syslogd: rsyslogd.

Este equipo es un router, por lo que no tiene manera de conectarle un teclado. Para peor, no tiene interfaz web ni telnet habilitado.

Pero puede controlarlo en forma remota mediante protocolo SNMP.

Desde la NMS, determine la dirección IP de este router y utilice los comandos snmpget / snmpset / snmptranslate para monitorear y/o configurar este dispositivo mediante las community 'public' y 'private'.

nms
Lab directory (host): /home/juanma/netkit/netkit-lab_snmpv2
Version: 1.2
Author: Mauro A. Meloni
Email: maurom@unlu.edu.ar
Web: https://github.com/redesunlu/netkit-labs/
Description:
A simple lab that enables a NMS to manage interface configuration of a remote router via SNMP v2

>>> End of host2 specific startup script.

host2
Lab directory (host): /home/juanma/netkit/netkit-lab_snmpv2
Version: 1.2
Author: Mauro A. Meloni
Email: maurom@unlu.edu.ar
Web: https://github.com/redesunlu/netkit-labs/
Description:
A simple lab that enables a NMS to manage interface configuration of a remote router via SNMP v2

--- Netkit phase 2 initialization terminated ---
```

Como se puede observar, el laboratorio está compuesto por 1 router, 2 hosts, y un NMS.

9. Busque, en los equipos del laboratorio y desde la estación de monitoreo **nms, que equipos tienen un servidor SNMP activo. Luego, descubra la dirección IP posee el router dentro de la red de gestión **10.0.0.0**.**

IP Host 1 = 198.51.100.1/24 → No se recibe respuesta.

IP Host 2 = 203.0.113.1/24 → No se recibe respuesta.

IP Router = 10.0.0.100/24 → Servidor SNMP activo.

IP NMS = 10.0.0.1/24 → Servidor SNMP activo.

10. Inicie una captura de tráfico en el enlace que une la entidad de gestión con el router (enlace M) mediante el comando **vdump M > captura-snmp.pcap. Mantenga la captura activa durante todos los ejercicios siguientes.**

Ahora, procedo a iniciar una captura en el enlace M (es decir, el enlace entre el Router y el NMS).

```
juanma@ubuntu:~/Desktop$ vdump M > captura-snmp.pcap
Running ==> uml_dump M
```

11. Busque en la documentación los OID que permiten obtener los siguientes datos. Utilizando el comando **snmpget desde la estación de monitoreo **nms** contra el router, determine:**

a. ¿Qué nombre y descripción posee el router?

```
root@nms:~# snmpget -v2c -c public 10.0.0.100 1.3.6.1.2.1.1.5.0
SNMPv2-MIB::sysName.0 = STRING: Router AYGR-11085
```

El nombre del router es: Router AYGR-11085

```
root@nms:~# snmpget -v2c -c public 10.0.0.100 1.3.6.1.2.1.1.1.0
SNMPv2-MIB::sysDescr.0 = STRING: Linux router 3.2.86-netkit-ng-K3.2 #2
Wed Mar 29 09:08:55 ART 2017 i686
```

La descripción es: Linux router 3.2.86-netkit-ng-K3.2 #2

b. ¿Cuántos datagramas ha recibido? ¿Cuántos datagramas ha reenviado?

```
root@nms:~# snmpget -v2c -c public 10.0.0.100 1.3.6.1.2.1.4.3.0
IP-MIB::ipInReceives.0 = Counter32: 10538
```

Datagramas recibidos: 10538

```
root@nms:~# snmpget -v2c -c public 10.0.0.100 1.3.6.1.2.1.4.6.0
IP-MIB::ipForwDatagrams.0 = Counter32: 10109
```

Datagramas reenviados: 10109

c. ¿Cuántas interfaces de red posee el router y en qué estado se encuentran? (Up/Down)

```
root@nms:~# snmpget -v2c -c public 10.0.0.100 1.3.6.1.2.1.2.1.0
IF-MIB::ifNumber.0 = INTEGER: 5
```

Cantidad de interfaces de red: 5

```
root@nms:~# snmptable -Cw 70 -v2c -c public 10.0.0.100 1.3.6.1.2.1.2.2 | less -S
SNMP table: IF-MIB::ifTable

  ifIndex ifDescr      ifType ifMtu ifSpeed  ifPhysAddress
    1      lo softwareLoopback 16436 10000000
    2     eth0 ethernetCsmacd 1500      0 4e:fb:86:4a:79:2b
    3     eth1 ethernetCsmacd 1500      0 ca:40:d5:e3:f9
    4     eth2 ethernetCsmacd 1500      0 ae:c0:25:dc:a9:90
    5     eth3 ethernetCsmacd 1500      0 ce:21:15:dd:d2:f6

SNMP table IF-MIB::ifTable, part 2

  ifAdminStatus ifOperStatus ifLastChange ifInOctets ifInUcastPkts
        up      up 0:0:00:00.00      200          4
        up      up 0:0:35:01.50  486286        6018
        up      up 0:0:00:00.00  477358        5895
        up      up 0:0:00:00.00   68333         829
        up      up 0:0:00:00.00           0           0
```

Estado de las interfaces: Las 5 interfaces están activas y operando.

d. Para cada uno de los puntos anteriores, indique el nombre del objeto que ha consultado, el OID correspondiente a cada uno, y el string de comunidad que ha utilizado.

a) Nombre y descripción del router:

Objeto consultado: iso.identified-organization.dod.internet.mgmt.mib-2.system.sysName

OID correspondiente: 1.3.6.1.2.1.1.5.0

Objeto consultado: iso.identified-organization.dod.internet.mgmt.mib-2.system.sysDescr

OID correspondiente: 1.3.6.1.2.1.1.1.0

b) Datagramas recibidos y reenviados:

Objeto consultado: iso.identified-organization.dod.internet.mgmt.mib-2.ip.ipInReceives

OID correspondiente: 1.3.6.1.2.1.4.3.0

Objeto consultado: iso.identified-organization.dod.internet.mgmt.mib-2.ip.ipForwDatagrams

OID correspondiente: 1.3.6.1.2.1.4.6.0

c) Cantidad de interfaces y estado de las mismas:

Objeto consultado: iso.identified-organization.dod.internet.mgmt.mib-2.interfaces.ifNumber

OID correspondiente: 1.3.6.1.2.1.2.1.0

Objeto consultado: iso.identified-organization.dod.internet.mgmt.mib-2.interfaces.ifTable

OID correspondiente: 1.3.6.1.2.1.2.2.0

12. Mediante el comando `snmpset`, deshabilite la interfaz número 1 (eth1) del router (pista: `ifAdminStatus.N`), para que no pueda haber comunicación posible con `host2`. Verifique que la interfaz está baja intentando hacer ping entre `host2` y `router`. Luego, utilizando el mismo comando, vuelva a habilitar la interfaz. Recuerde que para este ejercicio debe utilizar una **community especial.**

Para realizar esto, debo utilizar el siguiente comando:

`snmpset -VERSION -c COMUNIDAD AGENTE OID TIPO VALOR`

En este caso:

`snmpset -v2c -c private 10.0.0.100 1.3.6.1.2.1.2.2.1.7.1 i 2`

siendo:

1.3.6.1.2.1.2.2.1.7 (el OID correspondiente a `ifAdminStatus` dentro de `ifTable.ifEntry`)

El 3 a la derecha del 7 indica el número de la interfaz que quiero deshabilitar (eth1).

i el tipo INTEGER.

2 el valor down (para desactivar la interfaz).

```
root@nms:~# snmpset -v2c -c private 10.0.0.100 1.3.6.1.2.1.2.2.1.7.3 i 2
IF-MIB::ifAdminStatus.3 = INTEGER: down(2)
```

Ahora, verifico que efectivamente se deshabilitó haciendo ping al router desde `host2`:

```
root@host2:~# ping 10.0.0.100
PING 10.0.0.100 (10.0.0.100) 56(84) bytes of data.
From 203.0.113.1 icmp_seq=1 Destination Host Unreachable
From 203.0.113.1 icmp_seq=2 Destination Host Unreachable
From 203.0.113.1 icmp_seq=3 Destination Host Unreachable
From 203.0.113.1 icmp_seq=4 Destination Host Unreachable
From 203.0.113.1 icmp_seq=5 Destination Host Unreachable
From 203.0.113.1 icmp_seq=6 Destination Host Unreachable
From 203.0.113.1 icmp_seq=7 Destination Host Unreachable
From 203.0.113.1 icmp_seq=8 Destination Host Unreachable
From 203.0.113.1 icmp_seq=9 Destination Host Unreachable
^C
--- 10.0.0.100 ping statistics ---
```

Por último, ejecuto el comando para volver a activar la interfaz:

```
root@nms:~# snmpset -v2c -c private 10.0.0.100 1.3.6.1.2.1.2.2.1.7.3 i 1
IF-MIB::ifAdminStatus.3 = INTEGER: up(1)
```

13. Detenga la captura que inició en el punto 10 e inicie una nueva captura mediante el comando `vdump M > snmptrap.pcap`.

Ahora, procedo a detener la captura:

```
juanma@ubuntu:~/Desktop$ vdump M > captura-snmppcap
Running ==> uml_dump M
^CCaught signal 2, cleaning up and exiting
```

E inicio una nueva captura:

```
juanma@ubuntu:~/Desktop$ vdump M > snmptrap.pcap
Running ==> uml_dump M
```

14. En la estación de monitoreo `nms`, Utilice el comando `nc` para iniciar un servidor UDP en escucha en el puerto correspondiente a la recepción de traps SNMP.

Para realizar esto, utilizo el comando:

`nc -l -u -p 162` (puerto correspondiente a la recepción de traps SNMP)

```
nms
root@nms:~# nc -l -u -p 162
```

15. En el router, pulse la tecla Enter para forzar la caída de un enlace. En la `nms`, aguarde a la recepción del Trap (lo detectará por la salida de caracteres extraños en la pantalla de la `nms`).

Ahora, dentro del router presiono enter para forzar la caída de un enlace.

Una vez apretado enter, recibí lo siguiente:

```
nms
root@nms:~# nc -l -u -p 162
0public$
    @[-0v+C]00
+    +0
+0
+0
0
+
+0
```

16. Detenga el servidor `nc` y guarde la captura iniciada en el punto 13.

Una vez hecho esto, procedo a apretar CTRL + C en ambas terminales (primero en la del `nms` para detener el servidor `nc`, y luego en la terminal propia para detener la captura iniciada en el punto 13).

Trabajo práctico

1. ¿Qué comandos y aplicaciones necesita para poder hacer una consulta SNMP? (paquete **snmp**).

Para operar correctamente, el protocolo SNMP tiene que estar compuesto de 2 elementos: el agente y el gestor. Es una arquitectura cliente-servidor donde el agente es el servidor y el gestor es el cliente (excepto en el caso de las traps).

El agente es un software que se tiene que estar ejecutando en cada dispositivo administrado (managed device), los cuales pueden ser Routers, Switches, UPS, etc., el cual permite que los managed devices sean monitoreados.

Los agentes mantienen lo que conocemos como MIB, la cual es una base de información gestionada que contiene datos (que pueden ser comunes entre los dispositivos o específicos para cada dispositivo).

Los comandos utilizados para permitir la comunicación entre usuario y managed device tienen el siguiente formato:

En caso de ser una petición de tipo GET o similar (snmpwalk, por ejemplo):

```
[COMANDO] [-VERSIÓN] [-c COMUNIDAD] [AGENTE] [OID]
```

En caso de ser una petición de tipo SET (o sea, quiero establecer un valor dentro de la MIB):

```
snmpset [-VERSIÓN] [-c COMUNIDAD] [AGENTE] [OID] [TIPO] [VALOR]
```

Donde:

COMANDO es el comando específico que determina lo que estoy solicitando, por ejemplo: snmpget, snmpwalk, snmpbulkwalk, etc.

VERSIÓN es la versión de SNMP, por ejemplo: v2c, v3.

COMUNIDAD es el nombre de la comunidad con la que se identificará el cliente, por ejemplo: public, private.

AGENTE es la dirección IP o nombre de host a quien se consultará.

OID es el objeto a consultar (o en blanco para solicitar todo), por ejemplo: 1.3.6.1.2.1.1.5.0 (sysName).

TIPO es el tipo de datos del objeto (s=string, i=integer, ...)

VALOR es el valor que se le va a asignar al OID previamente nombrado.

Además de esto, se necesita que haya un puerto UDP abierto para la comunicación.

Por defecto, los puertos bien conocidos para SNMP son 161 (SNMP) y 162 (SNMP – Traps).

En cuanto a la información que brinda un agente:

a. ¿Cómo es posible conocer toda su información pública? Especifique los comandos que se deberían utilizar utilizados.

Es posible conocer toda la información pública mediante el uso de la comunidad "public", es decir, con el parámetro -c public indico que me estoy autenticando como miembro de dicha comunidad.

La comunidad viaja en texto plano y funciona como usuario y contraseña a la vez.

A su vez, cada comunidad puede tener permisos de lectura y escritura sobre cada OID.

Por ejemplo, en la siguiente imagen se detalla la diferencia entre utilizar el comando "snmpset" utilizando -c public y -c private:

```
nms
root@nms:~# snmpset -v2c -c public 10.0.0.100 1.3.6.1.2.1.2.2.1.7.3 i 1
Error in packet.
Reason: noAccess
Failed object: IF-MIB::ifAdminStatus.3

root@nms:~# snmpset -v2c -c private 10.0.0.100 1.3.6.1.2.1.2.2.1.7.3 i 1
IF-MIB::ifAdminStatus.3 = INTEGER: up(1)
```

Se puede ver que, al autenticarse como public, la ejecución del comando no arroja resultados favorables.

b. Como es posible saber que versiones de SNMP soporta el agente.

Para visualizar esto, utilicé el comando:

```
snmpstatus -V [AGENTE]
```

En este caso,

```
snmpstatus -V 201.148.106.3 (por ejemplo).
```

```
juanma@ubuntu:~/netkit/netkit-lab_snmpv2$ snmpstatus -V 201.148.106.3
NET-SNMP version: 5.8
```

Una vez visualizada la versión, procedo a buscar en la [Página oficial de Net-SNMP](#).

En este caso, esta versión del agente soporta tanto la versión 1 (SNMP v1) como la versión v2c (SNMP v2c) y la versión 3 (SNMP v3) utilizando tanto ipv4 como ipv6.

2. ¿Qué características tienen las OID? ¿Qué es un MIB?

OID:

Las OID tienen la característica de estar formadas por elementos separados por puntos. Estos elementos pueden ser nombrados o simplemente se puede hacer referencia a ellos mediante un número.

Se puede definir a una OID como una secuencia de números que se asignan jerárquicamente y cuya función es identificar objetos en la red.

Los OID son nodos dentro de la MIB, y existen 2 tipos:

1. Estructurales: son ramas, solamente tienen descrita su posición en el árbol.
Por ejemplo: IP → 1.3.6.1.2.1.4
2. De información: son hojas, son los nodos que se busca consultar y/o establecer valores.
Por ejemplo: sysName → 1.3.6.1.2.1.1.5

La definición de un OID contiene las siguientes partes:

1. El nombre del objeto.
2. El tipo del objeto.
3. Tipo de dato y valores posibles (por ejemplo: forwarding (1). not-forwarding(2)).
4. Tipo de acceso (por ejemplo: read, read-write).
5. Estado (Por ejemplo: obsolete, mandatory).
6. Descripción.
7. Nodo padre y número de hoja que representa el objeto actual (Por ejemplo: El OID de ipForwarding es 1.3.6.1.2.1.4.1, en este caso sería algo como ::= { ip 1 } ya que es la hoja n° 1 dentro del OID de IP).

MIB:

La MIB es la base de datos que contiene los nodos a los que se hace referencia mediante la OID.

Es una estructura jerárquica en forma de árbol que contiene información de todos los dispositivos gestionados en una red. Esta estructura define las variables que son utilizadas por el protocolo SNMP.

También puede ser definida como una colección de objetos identificados para la gestión, sus tipos y relaciones en una entidad gestionada.

3. El agente que trabaja con la versión 2 o 2c del protocolo, ¿Brinda solo información a la comunidad pública? ¿Cómo es posible saberlo? Describa, con un alto nivel de abstracción, los pasos necesarios para poder acceder a dicha información.

Una de las principales diferencias entre la versión 3 del protocolo SNMP (SNMPv3) y las versiones anteriores (SNMPv2, SNMPv2c y SNMPv1) es que la última versión tuvo como principal objetivo mejorar las características de seguridad del protocolo, implementando cifrado.

En la versión 2 o 2c, toda la información tanto de los mensajes SNMP como de la comunidad en si misma viaja como texto plano, por lo que puede ser interceptada por cualquiera que esté sniffendo la red.

Esto es crítico, ya que un atacante incluso podría intentar probar ejecutar distintos comandos snmp con las comunidades por defecto (public – private) o incluso alguna propia de algún fabricante específico.

Esto hace de SNMPv3 una versión con acceso seguro provisto por la autenticación (algoritmos MD5 o SHA) y el cifrado de paquetes a través de la red, el cual se realiza mediante el estándar de cifrado de datos (DES) o el estándar de cifrado avanzado (AES).

Para poder acceder a la información del protocolo SNMP (en versiones anteriores a SNMPv3) en una red (desde el lado del atacante) se podría sniffear la misma esperando que ocurra algún mensaje SNMP y luego, mediante la herramienta Wireshark (por ejemplo) analizar las PDUs de SNMP las cuales contienen dentro de “variable-bindings” toda la información del mensaje, desde el OID correspondiente hasta el valor del mismo.

4. A partir de la captura realizada [captura-snmp.pcap](#), elija un mensaje SNMP en particular y describa la PDU ejemplificando con los datos de la misma. Grafique además un esquema en el que identifique los equipos involucrados y sus roles desde el punto de vista del protocolo SNMP.

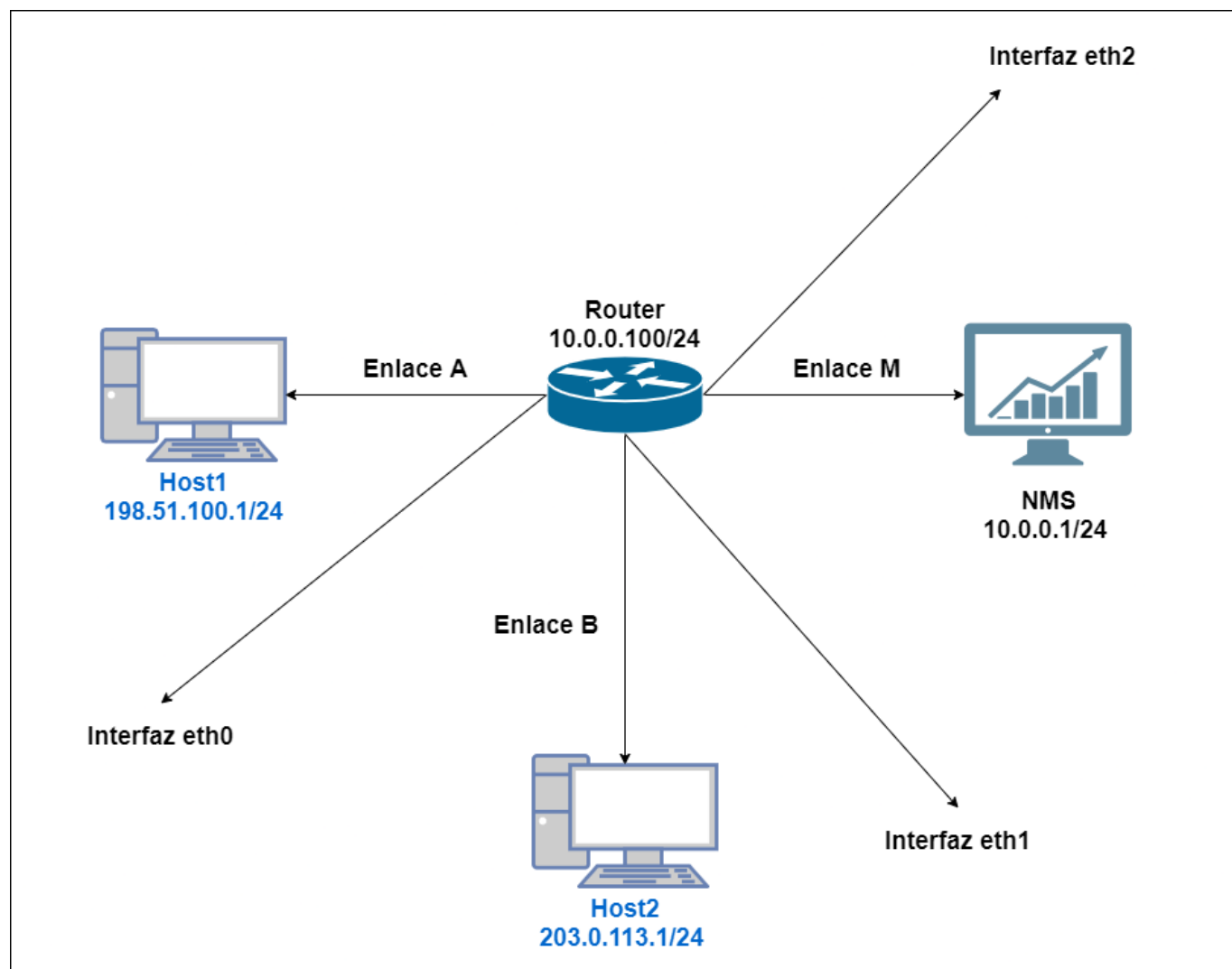
En este caso, opté por elegir el primer mensaje SNMP:

```
▼ Simple Network Management Protocol
  version: v2c (1)
  community: public
  ▼ data: get-request (0)
    ▼ get-request
      request-id: 1376832950
      error-status: noError (0)
      error-index: 0
      ▼ variable-bindings: 1 item
        ▼ 1.3.6.1.2.1.1.5.0: Value (Null)
          Object Name: 1.3.6.1.2.1.1.5.0 (iso.3.6.1.2.1.1.5.0)
          Value (Null)
```


PDU de mensaje SNMP N° 1

version	community	
v2c (1)	public	SNMP PDU
PDU Type		SNMP PDU
request-id		get-request (0)
error-status		1376832950
error-index		noError (0)
variable-bindings		0
		Object Name: 1.3.6.1.2.1.1.5.0 (iso.3.6.1.2.1.1.5.0)
		Value (null)

Los principales equipos involucrados son los siguientes:



Desde el punto de vista del protocolo SNMP, el NMS es quien gestiona los managed devices (en este caso el Router), y a su vez, recibe las traps de dicho dispositivo.

El router actúa como servidor SNMP, el cual recibe las peticiones (tales como mensajes snmpget, snmpset, etc.) de la NMS (cliente) por el puerto UDP 161.

Además de esto, el NMS también actúa como servidor ya que, recibe las traps enviadas desde los managed devices (cliente) por el puerto UDP 162.

5. Analizar la captura realizada en la experiencia de laboratorio **snmpwalk.pcapng** y **snmpbulkwalk.pcapng** ¿Qué diferencias se observan entre ambas capturas? ¿Qué implicancias tiene cada uno respecto al tráfico en la red y cómo se explica la diferencia en el tiempo de ejecución?

Para empezar a explicar las diferencias, primero voy a adjuntar los resultados de las capturas:

Snmpwalk:

1	0.000000000	192.168.66.131	198.12.32.85	SNMP	83	get-next-request	1.3.6.1.2.1.1
2	0.042640823	198.12.32.85	192.168.66.131	SNMP	104	get-response	1.3.6.1.2.1.1.1.0
3	0.042916359	192.168.66.131	198.12.32.85	SNMP	85	get-next-request	1.3.6.1.2.1.1.1.0
4	0.095617415	198.12.32.85	192.168.66.131	SNMP	93	get-response	1.3.6.1.2.1.1.2.0
5	0.096058068	192.168.66.131	198.12.32.85	SNMP	85	get-next-request	1.3.6.1.2.1.1.2.0
6	0.165161435	198.12.32.85	192.168.66.131	SNMP	89	get-response	1.3.6.1.2.1.1.3.0
7	0.165930054	192.168.66.131	198.12.32.85	SNMP	85	get-next-request	1.3.6.1.2.1.1.3.0
8	0.200001224	198.12.32.85	192.168.66.131	SNMP	85	get-response	1.3.6.1.2.1.1.4.0
9	0.201069836	192.168.66.131	198.12.32.85	SNMP	85	get-next-request	1.3.6.1.2.1.1.4.0

Snmpbulkwalk:

5	12.300652499	192.168.66.131	198.12.32.85	SNMP	83	getBulkRequest	1.3.6.1.2.1.1
6	12.340424932	198.12.32.85	192.168.66.131	SNMP	261	get-response	1.3.6.1.2.1.1.1.0 1.3.6.1.2.1.1.2.0 1.3.6.1.2.1.1.3.0 1.3.6.1.2.1.1.4.0

```

variable-bindings: 10 items
  1.3.6.1.2.1.1.1.0: 526f757465724f532052423330313155694153
  1.3.6.1.2.1.1.2.0: 1.3.6.1.4.1.14988.1 (iso.3.6.1.4.1.14988.1)
  1.3.6.1.2.1.1.3.0: 28834400
  1.3.6.1.2.1.1.4.0: <MISSING>
  1.3.6.1.2.1.1.5.0: 4d696b726f54696b
  1.3.6.1.2.1.1.6.0: <MISSING>
  1.3.6.1.2.1.1.7.0: 78
  1.3.6.1.2.1.2.1.0: 14
  1.3.6.1.2.1.2.2.1.1.1: 1
  1.3.6.1.2.1.2.2.1.1.2: 2

```

La principal diferencia que se puede apreciar es que, mientras el comando snmpwalk arroja todas las consultas y las respuestas por separado, snmpbulkwalk realiza la consulta y devuelve el resultado en una sola trama.

El modo de operar de ambos comandos es diferente:

Snmpwalk utiliza el get-response para obtener la respuesta y luego utiliza get-next-request para moverse entre diferentes hermanos de la misma rama, en cambio, snmpbulkwalk utiliza getBulkRequest para realizar la petición y con un simple y único get-response obtiene la lista de hojas de la rama solicitada.

La diferencia de tráfico en la red es notable ya que, mientras el mensaje **snmpwalk** genera 16 tramas (en el ejemplo de la captura), el mensaje **snmpbulkwalk** genera tan solo 2 tramas (las cuales contienen la misma información pero toda compactada en la trama de respuesta).

6. ¿Qué necesita una estación de monitoreo para poder recibir y procesar un Trap SNMP? ¿Es prudente cambiar el puerto en el que se reciben las traps?

Para recibir un TRAP SNMP, la estación de monitoreo debe tener un puerto UDP abierto.

Por defecto, el puerto por el cual se reciben las traps es el puerto UDP 162.

Según mi opinión, no es prudente cambiar el puerto en el que se reciben ya que todos los NMS tienen configurado dicho puerto por defecto, aunque podría ser modificado siempre y cuando se pueda configurar y hacer referencia al nuevo puerto elegido dentro de la NMS.

7. Analizar la captura `snmptrap.pcapng` (puede configurar Wireshark para habilitar la resolución de OID a nombre en el menú Edit » Preferences » Name Resolution » Enable OID Resolution; luego reinicie Wireshark para que cargue las MIBs existentes).

A partir de la PDU correspondiente al TRAP indique: ¿qué protocolo utiliza en cada capa OSI? ¿a qué evento corresponde la trap? ¿qué información se incluye?

La trap utiliza los siguientes protocolos por cada capa del modelo OSI:

7. En la capa de Aplicación, la trap utiliza el protocolo SNMP.

4. En la capa de Transporte, la trap utiliza el protocolo UDP.

3. En la capa de Red, la trap utiliza el protocolo IPv4.

2. En la capa de Enlace, la trap utiliza el protocolo Ethernet

El evento al que corresponde dicha trap es a la caída del enlace provocada al presionar enter dentro de la terminal del Router.

La información que trae es la siguiente:

1.3.6.1.2.1.1.3.0: 24022

El OID 1.3.6.1.2.1.1.3.0 corresponde a iso.identified-organization.dod.internet.mgmt.mib-2.system.sysUpTime.sysUpTimeInstance, cuyo valor es 24022.

1.3.6.1.6.3.1.1.4.1.0: 1.3.6.1.6.3.1.1.5.3

El OID 1.3.6.1.6.3.1.1.4.1.0 corresponde a iso.identified-organization.dod.internet.snmpV2.snmpModules.snmpMIB.snmpMIBObjects.snmpTrap.snmpTrapOID el cual contiene en su valor el OID de la trap, en este caso 1.3.6.1.6.3.1.1.5.3 la cual corresponde a iso.identified-organization.dod.internet.snmpV2.snmpModules.snmpMIB.snmpMIBObjects.snmpTraps.linkDown, o sea, el OID correspondiente a la caída de un enlace.

1.3.6.1.2.1.2.2.1.1.2: 2

El OID 1.3.6.1.2.1.2.2.1.1.2 corresponde a iso.identified-organization.dod.internet.mgmt.mib-2.interfaces.ifTable.ifEntry.ifIndex.

1.3.6.1.2.1.2.2.1.7.2: 2

El OID 1.3.6.1.2.1.2.2.1.7.2 corresponde a iso.identified-organization.dod.internet.mgmt.mib-2.interfaces.ifTable.ifEntry.ifAdminStatus, y su valor 2 indica que la interfaz está en estado “down” (igual que como se configuró mediante el ejercicio donde se utilizó el comando snmpset).

1.3.6.1.2.1.2.2.1.8.2: 2

El OID 1.3.6.1.2.1.2.2.1.8.2 corresponde a iso.identified-organization.dod.internet.mgmt.mib-2.interfaces.ifTable.ifEntry.ifOperStatus, y su valor 2 indica que la interfaz no está operativa (el OID anterior indicaba que estaba deshabilitada, este OID indica que no está “conectada”, es decir, que no se forma un enlace).

1.3.6.1.6.3.1.1.4.3.0: 1.3.6.1.4.1.8072.3.2.10

El OID 1.3.6.1.6.3.1.1.4.3.0 corresponde a iso.identified-organization.dod.internet.snmpV2.snmpModules.snmpMIB.snmpMIBObjects.snmpTrap.snmpTrapEnterp

rise, el cual indica la identificación autoritativa de la empresa asociada con la trap que se está enviando actualmente.

En este caso, su valor es 1.3.6.1.4.1.8072.3.2.10, el cual corresponde a iso.identified-organization.dod.internet.private.enterprise.net-snmp.netSnmpEnumerations.netSnmpAgentOIDs.linux, es decir, el OID correspondiente al agente net-snmp para el sistema operativo Linux.