



## **Project 1 - Using SNMP tools**

### **MANAGING AND SECURING COMPUTER NETWORKS**

---

**Floriane Magera** (S111295)  
**Fabrice Servais** (S111093)

March 10, 2015

## 1 Retrieving variables manually

The relevant MIB files according to the questions are the following:

- SNMPv2-MIB (sysDescr)
- IP-MIB (ipForwarding, ipDefaultTTL)
- IF-MIB (ifNumber, ifDescr, ifType, ifOperStatus, ifPhysAddress, ifMtu)

Those three MIB's have been added in the `/.snmp/snmp.conf` file.

### 1.1 What is the system description (sysDescr)?

We use the `snmpget` command which will get the value of the leaf `sysDescr` (1.3.6.1.2.1.1.0) on the OID tree.

```
snmpget hawk.run.montefiore.ulg.ac.be sysDescr.0 -v 2c -c run69Zork!
```

This is the reply:

```
SNMPv2-MIB::sysDescr.0 = STRING: Cisco Internetwork Operating System Software
IOS (tm) C2600 Software (C2600-IK803S-M), Version 12.2(11)T, RELEASE SOFTWARE
(fc1) TAC Support: http://www.cisco.com/tac Copyright (c) 1986-2002 by cisco
Systems, Inc. Compiled Thu 01-Aug-02 12:47 by cca"
```

### 1.2 Is IP forwarding enabled?

In the same way we used `snmpget` previously, we get the value of the object `ipForwarding` (1.3.6.1.2.1.4.1.0).

```
snmpget hawk.run.montefiore.ulg.ac.be ipForwarding.0 -v 2c -c run69Zork!
```

This is the reply:

```
IP-MIB::ipForwarding.0 = INTEGER: forwarding(1)
```

The value is set to "forwarding".

### 1.3 How many interfaces are present in that router?

With the same command, we find the value of `ifNumber` (1.3.6.1.2.1.2.1.0):

```
snmpget hawk.run.montefiore.ulg.ac.be ifNumber.0 -v 2c -c run69Zork!
```

This is the reply:

```
IF-MIB::ifNumber.0 = INTEGER: 5
```

There are so 5 interfaces.

To get the informations about each entry, we use the `snmpwalk` command which will get the subtree from the OID node that is given to it, in our case `ifTable` (1.3.6.1.2.1.2.2) :

```
snmpwalk hawk.run.montefiore.ulg.ac.be ifTable -v 2c -c run69Zork!
```

We got all the informations contained in that table, here are the useful ones :

```
IF-MIB::ifIndex.1 = INTEGER: 1
IF-MIB::ifIndex.2 = INTEGER: 2
IF-MIB::ifIndex.3 = INTEGER: 3
IF-MIB::ifIndex.4 = INTEGER: 4
IF-MIB::ifIndex.5 = INTEGER: 5
IF-MIB::ifDescr.1 = STRING: FastEthernet0/0
IF-MIB::ifDescr.2 = STRING: Serial0/0
IF-MIB::ifDescr.3 = STRING: FastEthernet0/1
IF-MIB::ifDescr.4 = STRING: Serial0/1
IF-MIB::ifDescr.5 = STRING: Null0
IF-MIB::ifType.1 = INTEGER: ethernetCsmacd(6)
IF-MIB::ifType.2 = INTEGER: propPointToPointSerial(22)
IF-MIB::ifType.3 = INTEGER: ethernetCsmacd(6)
IF-MIB::ifType.4 = INTEGER: propPointToPointSerial(22)
IF-MIB::ifType.5 = INTEGER: other(1)
IF-MIB::ifMtu.1 = INTEGER: 1500
IF-MIB::ifMtu.2 = INTEGER: 1500
IF-MIB::ifMtu.3 = INTEGER: 1500
IF-MIB::ifMtu.4 = INTEGER: 1500
IF-MIB::ifMtu.5 = INTEGER: 1500
IF-MIB::ifPhysAddress.1 = STRING: 0:7:85:a8:83:20
IF-MIB::ifPhysAddress.2 = STRING:
IF-MIB::ifPhysAddress.3 = STRING: 0:7:85:a8:83:21
IF-MIB::ifPhysAddress.4 = STRING:
IF-MIB::ifPhysAddress.5 = STRING:
IF-MIB::ifOperStatus.1 = INTEGER: up(1)
IF-MIB::ifOperStatus.2 = INTEGER: up(1)
IF-MIB::ifOperStatus.3 = INTEGER: up(1)
IF-MIB::ifOperStatus.4 = INTEGER: down(2)
IF-MIB::ifOperStatus.5 = INTEGER: up(1)
```

For each interface, we have its index (`IF-MIB::ifIndex`), its name (`IF-MIB::ifDescr`), its type (`IF-MIB::ifType`) and its status (`IF-MIB::ifOperStatus`). For the latter, the value is set to "up" when the interface is on and "down" when the interface is off.

## 1.4 What is the MAC address of FastEthernet0/0?

We saw that "FastEthernet0/0" is at the index 1 in the table. Knowing that the MAC address is contained in `ifPhysAddress` (OID: 1.3.6.1.2.1.2.2.1.6), we can access the information using:

```
snmpget hawk.run.montefiore.ulg.ac.be ifPhysAddress.1 -v 2c -c run69Zork!
```

This is the reply:

```
IF-MIB::ifPhysAddress.1 = STRING: 0:7:85:a8:83:20
```

## 1.5 What is the MTU of Serial0/1?

We saw that "Serial0/1" is at index 4 in the table. Knowing that the MTU is contained in `ifMtu` (OID: 1.3.6.1.2.1.2.2.1.4), we can access the information using:

```
snmpget hawk.run.montefiore.ulg.ac.be ifMtu.4 -v 2c -c run69Zork!
```

This is the reply:

```
IF-MIB::ifMtu.4 = INTEGER: 1500
```

## 1.6 What is the default IP TTL?

The information is contained in the `ipDefaultTTL` (OID: 1.3.6.1.2.1.4.2).

```
snmpget hawk.run.montefiore.ulg.ac.be ipDefaultTTL.0 -v 2c -c run69Zork!
```

```
IP-MIB::ipDefaultTTL.0 = INTEGER: 255
```

The TTL is so of 255.

## 1.7 Set the default IP TTL to a different value

We use the `snmpset` command to try to set a value into the `ipDefaultTTL` field.

```
snmpset -v 2c -c run69Zork! hawk.run.montefiore.ulg.ac.be ipDefaultTTL.0 i 200
```

This is the reply:

```
Error in packet.
```

```
Reason: noAccess
```

```
Failed object: IP-MIB::ipDefaultTTL.0
```

```
zsh: exit 2      snmpset -v 2c -c run69Zork! hawk.run.montefiore.ulg.ac.be ipDefaultTTL.0 i 200
```

We have thus no write access to the object. Since the MIB access policy for that object is "read-write", it means that the SNMP mode access (related to the community) is set to "READ-ONLY".

## 2 Retrieving variables from a script

In order to count the number of packets incoming and outgoing from Hawk, we based our estimations on the IP counters. We neglect the packets which are not datagrams. Most of the objects used are deprecated but as the alternatives recommended by Cisco were not available on Hawk, we were forced to use it.

For the incoming packets, we used `ipInReceives` (1.3.6.1.2.1.4.3) which counts all the packets received.

For the outgoing packets, we used several objects.

- **ipOutRequests** (1.3.6.1.2.1.4.10) : the number of packets submitted by higher layers for transmission, it does not count the forwarded packets.
- **ipOutDiscards** (1.3.6.1.2.1.4.11) : the number of packets discarded.
- **ipOutNoRoutes** (1.3.6.1.2.1.4.12) : the number of packets discarded because there was no known route to their destination.
- **ipForwDatagrams** (1.3.6.1.2.1.4.6) : the number of packets forwarded.

We realized too late that the **ipOutDiscards** and **ipOutNoRoutes** are already included in **ipForwDatagrams**. There is obviously an impact on our collected data, even if the number of packets discarded or for which no route was found is surely not very high.

We notice on the graph two peaks in the number of packets, the higher one around 9:55 AM and the second one around 12:25 AM.

