NMIS Handling Counter64 using SNMPv1

Background

SNMP is a protocol specification which is standardised in the IETF. SNMPv1 does not support 64 bit counters; 64 bit counters were one of the primary drivers behind the development of SNMPv2 and ultimately SNMPv2c. SNMPv1 was ratified in 1990 with RFC 1155, RFC 1156 and RFC 1157, the MIB definitions were updated in RFC 1213. SNMPv2c was ratified in 1996, which makes 2016 its 20th birthday.

Why are 64 bit counters important?

Cisco published a great article on this: SNMP Counters: Frequently Asked Questions, and RFC 2233 summarises the issue best:

```
"As the speed of network media increase, the minimum time in which
a 32 bit counter will wrap decreases.  For example, a 10Mbs stream
of back-to-back, full-size packets causes ifInOctets to wrap in
just over 57 minutes; at 100Mbs, the minimum wrap time is 5.7
minutes, and at 1Gbs, the minimum is 34 seconds.  Requiring that
interfaces be polled frequently enough not to miss a counter wrap
is increasingly problematic."
```

The Problem

Despite this interesting bit of history, some network device vendors still only support SNMPv1, which isn't a problem unless you have high speed interfaces (for example 1 gigabit per second, which some of these vendors *do* have). Funnily enough these same vendor companies have only existed for 10-15 years, so are younger than the standards they should be implementing.

In the other corner we have NMIS, which is Open Source and - by the nature of Open Source and of software relying on IETF protocols and specifications - strongly standards-oriented.

Making NMIS work with SNMPv1 using 64-bit counters

NMIS itself is not strongly typed and doesn't have a problem with 64-bit counters; the issue here is the set of SNMP libraries that NMIS uses, which do strictly enforce the standards.

How will I know that I need to change NMIS?

NMIS will be reporting an error for SNMP collection operations; the message will be something like "The Counter64 type is not supported in SNMPv1". This would be seen in the debug output and also in the NMIS Event log. For example:

```
19:25:13 checkResult, SNMP ERROR (Ubiquiti-AirFiber1) (1.3.6.1.4.1.41112.1.3.3.1.1) The Counter64 type is not
supported in SNMPv1
```

If you see this, then you have a device that is cheating by using a non-standard protocol-datatype combination.

What do I have to change?

To make NMIS work with this non-standard combination, you need to modify the SNMP library code to prevent it from detecting (and aborting on) the datatype mismatch.
NMIS uses the Net::SNMP perl module and only one file belonging to this library needs to be adjusted. The file in question is `Message.pm` (or in Perl terms the module `Net::SNMP::Message`);this is the object which manages the SNMP messages.
First locate `Message.pm` on your system, which will normally be in one of the perl directories under /usr or /usr/local.
Simply use the Unix `find` command to do the lookup work for you:

```
find /usr/ -type f -path "*/Net/SNMP/Message.pm"
```

The result should be similar to this:

```
$ find /usr/ -type f -path "*/Net/SNMP/Message.pm"
/usr/share/perl5/Net/SNMP/Message.pm
/usr/local/share/perl5/Net/SNMP/Message.pm
```

If `find` returns more than one match, run `perl -V` and check the `@INC` list: the module files are looked up in all `@INC` directories in order, and the first match wins. That's the file you want to modify. In the example above, the module in `/usr/local/share` wins over the one in `/usr/share`.

- First, make a backup (as the `root` user): `cp /usr/local/share/perl5/Net/SNMP/Message.pm /usr/local/share/perl5/Net/SNMP/Message.pm.unpatched`
- Next, edit the file `/usr/local/share/perl5/Net/SNMP/Message.pm` (again as the `root` user) and look for '`sub _process_counter64`'. In version 3.0.1 it is on line 1650 of that file. You want to disable the `if` statement which generates the error and returns; just comment out the 'problematic' code with '#'.
  It is also advisable to add a comment that explains why the change wa made; In the example below, the editor was started with line number display, lines 1655 to 1657 have been commented out,
  and line 1654 was amended to describe the change.

```
1650 sub _process_counter64
1651 {
1652    my ($this, $type) = @_;
1653
1654    # Verify the SNMP version - disabled for non-standard V1+64bit combination
1655    #if ($this->{_version} == SNMP_VERSION_1) {
1656    #   return $this->_error('The Counter64 type is not supported in SNMPv1');
1657    #}
1658
1659    # Decode the length
```

- Finally, run an NMIS update on the node in question; this will verify that your modified code still compiles and runs, and that NMIS works around the non-standard device behaviour.
  Should you get weird Perl errors, restore your backup and try the edit operation again.

Please note that if the perl library is updated (by `apt-get` or `yum` or some other software administration tool), then your adjustment will likely be lost and you will need to do it again.