NeTraMet 4.1 Users' Guide

NeTraMet Release Notes

Version 4.1

Nevil Brownlee

Information Technology Systems & Services
The University of Auckland
Auckland, New Zealand

November 1997

1. Introduction

Version 4 of NeTraMet introduces many changes.

The most significant of these result from implementing the RTFM Meter MIB, which requires the features of SNMPv2; in return meters can now run multiple rule sets, which greatly increases their usefulness. In addition, 64-bit counters are used for packet and byte counts, greatly reducing the chance of counters rolling over between meter readings. Many of the differences between v4 and earlier versions result from these changes.

Other features, such as the ability to use IP ports other than 161 (SNMP) and NeMaC's improved checking of rule sets, were prompted by operational needs. Still others have been suggested by NeTraMet users, such as the ability to meter four Ethernet interfaces on a PC.

In spite of these changes, the new version is compatible with version 3, in that version 3 meters continue to work properly with version 4 managers. Of course version 3 meters can't run multiple rule sets, that requires changing to version 4 meters.

Version 4.1 beta 1 was released in May 97. It had reached beta 15 before the 'official' 4.1.0 release in Nov 97. It has been in production use for many months at many sites around the world. Thanks to all those who have provided - and continue to provide - feedback to guide NeTraMet's development.

1.1. Other Documentation: RFCs

So far there are four RFCs which are of direct interest to NeTraMet users. They are:

RFC 1272 "Internet Accounting: Background"
D. Hirsh, C. Mills, G. Ruth, Nov 1991

Explains the model for an Internet Accounting system, with detailed discussion of the issues to be considered.

RFC 2063 "Traffic Flow Measurement: Architecture",
N. Brownlee, C. Mills, G. Ruth, Jan 1997

Describes in detail the RTFM (Realtime Traffic Flow Measurement) architecture. This is an architectural model for a network accounting system.

RFC 2064 "Traffic Flow Measurement: Meter MIB",
N. Brownlee, Jan 1997

RFC 2123 N. Brownlee, "Traffic Flow Measurement: Experiences with NeTraMet", Mar 1997

2.1. 'NoMatch' instead of 'Retry'

'nomatch' is now a synonym for 'retry' as an action in rules. This name was discussed at the Montreal RTFM WG session, and is used in the ruleset examples given in RFC 2123. You should use 'nomatch' in all new rulesets.

2.2. 'MatchingStoD' Attribute

The 'MatchingStoD' attribute has been added. This is a Packet Matching Engine (PME) attribute indicating the order in which a packet's addresses are being matched. Its value is 1 if the addresses are in 'StoD' order (i.e. as they appear 'on the wire'), and 0 if the packet is being matched with its addresses swapped. It was discussed and agreed to at the San Jose meeting in 1996.

It can be used to perform different actions when the match is retried, thereby simplifying some kinds of rule sets. For example, consider a ruleset which classifies packets for a list of networks - the 'usual' networks - and counts packets and bytes for them. For 'unusual' packets (those from networks not in the list) we would like to save the source and destination PeerAddresses.

To count such an 'unusual' packets we need to know the matching direction: the MatchingStoD attribute provides this. To use it, one follows the source address tests with a rule which tests whether the matching direction is S->D (MatchingStoD value is 1). If so, a 'NoMatch' action can be executed. Otherwise, the packet has failed to match in both directions; one can Push whatever attribute values are of interest and count the 'unusual' packet.

2.3. 'meter IP port' option (meters and managers)

NeTraMet meters are SMP or SNMPv2 agents. By default they expect to use IP port 161 (the SNMP port) to send and receive packets from a manager such as NeMaC, nm_rc or nifty. In most cases this works well, especially when the NeTraMet meter is running on a PC. Problems can arise, however, when it is running on a Unix system.

The most common situation occurs when the Unix system is already be running an SNMP daemon, which uses port 161. If this can be killed off, NeTraMet can be run normally. If it is being used for system monitoring, NeTraMet will have to use another port.

From version 4.1 all the NeTraMet meters and managers allow the user to specify the IP port they are to use, using the -l command-line option. This takes the form

```
-m pp
```

where pp is the port number to uses. In most cases it will be sensible to use a port number above 2048, so as not to conflict with the well-known ports.

2.4. 'length from IP headers' option (meters only)

Byte counts in NeTraMet are normally computed using the lengths of packets as they appear 'on the wire,' i.e. as reported by the interface hardware driver. The

```
-l
```

NeTraMet meter option specifies that meter should use the length field from IP headers instead. The resulting byte counts indicate the IP data bytes, rather than the actual bytes on the wire.

2.5. Syntax Checking of Rule Sets

Before NeMaC downloads a rule file to a meter, it performs as much checking as it can. To do this NeMaC has a three-pass assembler for the ruleset. Pass one verifies that sensible values are given for each of the five fields of each rule, and builds a symbol table (symbols can only be used as labels for rules). Pass two associates gotos with their targets, and pass three downloads the rules to the meter.

NeMaC version 4.1 adds more checking. It attempts to analyse control flow through the ruleset, so as to catch common errors such as Gotos which should have been GotoActs.

The new error messages, alas, may be confusing. They are:

```
Possible flow into rule: must GotoAct
```

```
Possible Goto to this rule: must GotoAct
```

This rule has an action of Assign, AssignAct, CountPkt, PushPkt or PushPkttoAct. These rules should be executed via a GotoAct.

Assign and AssignAct use the Attribute field as an argument, so it doesn't make sense to perform the test.

CountPkt, PushPkt and PushPkttoAct all push a value from the packet, so it's not sensible to use the Value field of the rule in a test. If you want to test a value, then push that value if the test succeeds, use the Push action!

```
Rule can't be executed
```

This rule cannot be executed either by a Goto or GotoAct, or by a flow-on from the previous rule. This is a warning message - it won't prevent NeMaC from downloading the ruleset. Sometimes it can be useful to leave such rules in a rule set - if this is the case, just ignore the message.

2.6. Multiple Interfaces on Meters

In version 3.4 multiple interfaces were only supported for Unix meters. Version 4.1 extends this capability to the PC meters. To use this on a PC, one has to install up to four Ethernet cards, each with its own hardware interrupt, and assign each of them a software interrupt number. The interfaces are configured using the following NeTraMet command-line options:

-i ifn Tells NeTraMet which interface to monitor. If this is not specified libpcap chooses the default Ethernet interface. Up to four interfaces may be specified for the Unix and PC meters. For the Unix meter, ifn is the interface name, e.g. le0. For the PC meter it is the software interrupt number in decimal, e.g. 96 = 0x60.

-h ifn As for -i (above), except that if you have a packet driver which implements the 'high-performance' driver specification, NeTraMet will take advantage of it.

-I ifn Tells PC NeTraMet to use the specified interface for IP communication, but not to monitor it. If this option is used, up to three other interfaces may be monitored. If no interface is specified as 'IP only,' the first interface appearing as a -i or -h option will be used as the meter's IP interface.

For example

```
meter_pc  -i120 -i121 -i122  -I123  -w test_meter
```

would meter traffic on interfaces 0x78, 0x79 and 0x7A, with SNMP traffic reaching the meter via interface 0x7B.

3. Meters and Multiple Rule Sets

The RFC 2064 Meter MIB allows the meter to run multiple rulesets. It does this in the same way as the RMON MIB - by having 'Information' tables for Rulesets, Tasks and Readers. These tables also carry an 'Owner' name, to identify which manager they belong to.

3.1. Meter Information Tables

To use a meter, a manager must create entries in the information tables. For example, to download a ruleset to a meter, the manager first creates a row in its Ruleset Info table. After that, the manager refers to that ruleset by its index in the Ruleset Info table.

The tables are:

Ruleset Info: One entry for each ruleset in the meter. Ruleset Info variables include the ruleset size, its 'name' (an integer for version 4.1), and the number of flows in the flow table which were observed by the ruleset.

Task Info: One entry for each task the meter is running. Task Info variables indicate 'current' and 'standby' ruleset indexes (in the Ruleset Info table), and HighWaterMark for this task. Note that each task may have a different HighWaterMark.

Reader Info: One entry for each meter reader which is collecting flow data from the meter. Reader Info variables include the times of the last two data collections, and a timeout value. If this time elapses without any collections the meter will delete this reader from the Reader Info table.

3.2. Owner Names

An 'Owner Name' is an alphanumeric identifier, up to 16 chars long, which is used to identify rulesets, tasks and meter readers in the meter control tables. It is specified as an extra (new) parameter on the manager command line, or (for NeMaC) on config file lines. If it is not specified, NeMaC, nm_rc and nifty use 'NeMaC,' 'nm_rc,' or 'nifty' as the Owner Name.

The managers differ slightly in the way they start and stop rulesets:

nm_rc and nifty begin operation by examining the meter's Info Tables and destroying any entries with the same Owner Name as themselves. When they themselves terminate normally (for example when the user presses the Control-C key), they destroy the ruleset they were using on the meter. This makes it simple to use nm_rc and nifty without an Owner Name, and know that they won't leave rules sets running on the meter. Any long-running instances of nm_rc or nifty should be run using Owner Names which indicate who they belong to.

NeMaC takes a different approach to running rule sets. If it is terminated by the user (using Control-C), it destroys all the rulesets it is running on all its meters. This is an unusual situation, since NeMaC is usually left running for months at a time. Should NeMaC fail, for example because its Unix host is rebooted, the user will normally want to retrieve flow data which has accumulated in the meter. This need is met by NeMaC's 'doWnload level' option:

-w nn nn = 0 (the default) downloads rules on meter startup and after a meter restart.

nn = 1 downloads only after a meter restart

nn = 2 never downloads.

For nn values 0 and 1, NeMaC destroys any rulesets running on the meter with NeMaC's Owner Name (deleting any flow data accumulated by them) before it downloads.

For production use, one should start NeMaC using nn = 0, and arrange for it to restart after system reboots using nn = 1. This should minimise the amount of flow data lost because of NeMaC restarts.

3.3. Manager Task Options

-E nn

This is a new NeMaC option which specifies the timeout (in seconds) for rEeader Info rows. If collections stop (e.g. because a manager has failed), the meter will delete the row after this time. The default is 0, i.e. the row will never time out.

-h pp

This NeMaC option specifies HighWaterMark for a manager task. In v3 the meter default was 65 (percent). In v4.1 the default is 0 (no test for high water).

3.4. Checking Meter Status

3.4.1. Meter Commands: U, A, E

The meter Information tables may be displayed at the meter using console commands:

U displays the rUleset Info table

A displays the tAsk Info table

E displays the rEader Info table

3.4.2. nm_st program

The nm_st program is a simplified form of RTFM manager which doesn't attempt to download rules or to collect flow data. It simply reads the information tables from a meter and displays them.

For example

```
nm_st  -c20  130.216.234.237 test
```

displays the ruleset, task and reader tables for meter 130.216.234.237, which has 'test' as its SNMP read community. nm_st also displays the processor utilisation of the meter; the -c20 option specifies that it should do this every 20 seconds.

4. Changing from v3 to v4

4.1. Meter - Manager Compatibility

Version 4 managers such as NeMaC will work properly with Version 3 meters. Version 3 managers, however, will NOT work with version 4 meters, since the version 3 manager cannot use the SNMPv2 commands which they require.

To change to using version 4 you should change your managers first, then your meters. If it is helpful, you can change the meters one at a time - the manager will see a meter (either version) restart, and will automatically download the rulesets to it.

4.2. Changes in Flow Data Files

Version 4 of NeTraMet allows the meter to run multiple rule sets. This means that the 'rule set number' attribute indicates a row in the meter's Rule Info table, which is chosen at random by NeMaC when it downloads a rule set. To allow Analysis Applications to associate the rule set numbers in a flow data file with the rule sets which produced them, a new information record has been added, the 'Ruleset' record. The format of this is

#Ruleset: *nn setname rfname owner*

The fields are

nn	ruleset number, as it appears in flow data records
setname	Name of the ruleset, from its SET statement. For v3 and v4.1 this can only be an integer
rfname	Name of the rule file, e.g. rules.x_ip
owner	Owner name for this rule set

Within flow data file there are two changes to the record formats:

- Dates now use four digits for the year (1997 instead of 97)
- The integer values used for PeerTypes have been changed to use the values specified in the 'Assigned Numbers' RFC (RFC 1700). You should not be affected by this unless you have analysis applications which use PeerTypes to distinguish flows.

The PeerType values are:

PeerType	*Version 4*	*Version 3*
IP4	1	2
IP6	2	
NSAP (CLNS)	3	3
Other	6	12
Novell (IPX)	11	6
EtherTalk	12	7
DECnet	13	5

Note: Although a PeerType value has been assigned for IP version 6, NeTraMet doesn't (yet) recognise IPv6 packets.

5. NeTraMet Meters on the PC

5.1. 32-bit and 16-bit PC Meters

Earlier versions of NeTraMet (the RTFM Traffic Meter) provided only a 16-bit PC DOS meter. Being a DOS program, this could only access the bottom 640 KB of PC memory, which limited it to a maximum of about 3,000 flows in its flow table.

The Version 4 development effort included porting NeTraMet to run as a 'statistics module' within the OC3MON program (an AM network monitor developed by NLANR and MCI), which runs on a PC with 128 MB (or more) of memory. This has allowed me to produce a 32-bit PC version of the NeTraMet meter.

The 32-bit meter uses all of the available memory. 16 MB of memory should allow it to handle a table of 100,000 flows or more..

The Version 4.1 distribution includes the 16-bit meter (ntm16.exe) and the 32-bit meter (ntm32.exe). These will run on a 386, 486 or Pentium PC. The PC source files provide makefiles for a Pentium-only version (ntm32p) and the OC3MON version (ntmoc3).

5.2. Installing a PC NeTraMet Meter

The ntm41-pc.zip file contains most - if not all - of the files you need to set up and run the 16- and 32-bit versions of the PC NeTraMet meter.

The 16-bit meter will run on a 386 with 640 kB of memory, the 32-bit meter requires at least a 386 with 4 MB of memory. If you are buying a new PC for use as a meter, I suggest a minimum of a 90 MHz Pentium with 8 MB of memory. You should use the 32-bit meter rather than the 16-bit one.

To install a PC meter, proceed as follows:

1) Format a high-density floppy disk as a system (bootable) disk.

   Unzip this file with the -d option onto the floppy disk; this will create directories called WINDOWS, DRIVERS and NETRAMET.

2) Edit the wattcp.cfg file so as to set the IP address, netmask and gateway addresses, and the domain name to those which are correct for your meter.

   Edit the pd.bat file so that it is correct for the Ethernet card(s) you are using. The DRIVERS directory contains packet drivers for NE2000, SMC_WD and 3C509 cards; these have been modified so as to support NeTraMet's 'high-performance packet driver' option. If you are using any other kind of Ethernet card you'll have to copy a packet driver for it into this directory.

3) NeTraMet can handle up to 4 Ethernet cards. You'll need a line for each card in the pd.bat file, with different software interrupts for each. I find interrupts 120, 121, 122 and 123 (decimal) work properly on most PCs. Note that the 32-bit version requires the packet driver(s) to be loaded in low memory. Loadhigh will NOT work!

4) If you will be using the 32-bit NeTraMet, you will need to copy two files from your meter PC's WINDOWS directory into the WINDOWS directory of the floppy. The files are HIMEM.SYS and EMM386.EXE. They are required to support the 32-bit environment for NeTraMet.

   Note that if you are using PCI cards you need EMM386 version 4.49 or later. You can find out the version number by typing EMM386 at the DOS prompt.

   If you will be using the 16-bit NeTraMet, you should comment out the lines in config.sys which refer to the WINDOWS files.

5) In the NETRAMET directory, edit the acct.bat file so as to specify the startup parameters. For example

```
ntm32  -f80000  -w write-com  -r read-com
```

Simplest case. Uses 32-bit meter on a single interface, with write community 'write-com' and read community 'read-com.' The maximum number of flows in the meter will be 80000. NB: you should NOT leave the read community as 'public' !

```
ntm16  -h120 -h121 -I122  -w write-com  -r read-com
```

Uses 16-bit meter to measure flows on high-performance-driver interfaces using interrupts 120 and 121, while using Ethernet 122 for IP communications with the meter. No metering is performed on interface 122. The default number of flows on the 16-bit meter is 2000.

5.3. Building the PC Meters

All source and make files for PC NeTraMet (including SNMPv2C, WATTCP and OC3MON) are contained in the ntm41src.zip file. Unzip these files using the -d option; this will produce the following directories:

SNMPLIB	*Source for the SNMPv2C library*
WATTCP	*Source for the Waterloo TCP (WATTCP) library*
NETRAMET	*NeTraMet meter source*
NTM16	*Make files for the 16-bit meter*
NTM32	*Make files for the 32-bit meter*
NTM32P	*Make files for the 32-bit Pentium-only meter*
NTMOC3	*Make files for the OC3MON meter*

Other directories contain source files for OC3MON, some of which are used to provide hardware support for the 32-bit versions of NeTraMet.

To build a meter, you need Borland C++ version 4.5, and Borland's PowerPack for DOS.(which provides a 32-bit DOS environment). Once you have unzipped the distribution files into the directories (as above), cd into the directory for the meter you want to build, e.g. NTM32 for the 32-bit meter. Now proceed as follows:

`make -fSNMP`	to build the SNMP library
`make -fWATTCP`	to build the WATTCP library
`make`	to build the meter

6. Building NeTraMet Unix Programs using Autoconfig

6.1. Meter - Manager Compatibility

The version 4 release of NeTraMet includes subdirectories containing Makefiles for various operating systems - exactly as for version 3 - and these can be used to build Unix meter and manager programs, as detailed in the NeTraMet Manual.

Version 4 also includes an autoconf directory, which contains a complete set of GNU Autoconfig files for building the NeTraMet programs on any Unix system. These use Autoconfig's WORDS_BIGENDIAN and SIZEOF_LONG defines to specify the system's byte ordering and word size. They are used to implement native Alpha code to get and put the values of 64-bit counters.

The recommended method of building NeTraMet is to use autoconfig. Details are given in the INSTALL file in the autoconf/ directory. Briefly, the process is as follows:

1) Install libpcap
   (The configure script needs to be able to find the libpcap.a library file.)
2) Set the LIBS environment variable to the directory containing libpcap:
   ```
   setenv LIBS -L/usr/local/lib
   ```
3) Make a directory for NeTraMet, e.g. ntm
   ```
   cd ntm
   ```
4) Copy the distribution file (NeTraMet.tar.gz) into the ntm directory.
5) Unzip it
   ```
   gzip -d NeTraMet.tar.gz
   ```
6) Change directory into the autoconf directory:
   ```
   cd autoconf
   ```
7) Make sure there is no old (cached) autoconfig material lying around:
   ```
   rm config.*
   ```
8) Run the configure script to build NeTraMet's make files:
   ```
   ./configure
   ```
10) Make NeTraMet:
   ```
   make
   ```

The autoconfigure process has been found to work correctly on Solaris, Irix, Linux and DEC Unix. It is now the preferred method of building NeTraMet. The old directories containing system-dependent Makefiles will be discontinued in later versions of NeTraMet.

7. Author's Address

Please send any comments, suggestions, bug reports to me, Nevil Brownlee, i.e.

n.brownlee@auckland.ac.nz