NAME

nafilter − NetSA Aggregated Flow filter and sorter

SYNOPSIS

```
nafilter    [--in INPUT_SPECIFIER] [--out OUTPUT_SPECIFIER]
            [--nextdir PROCESSED_INPUT_DIRECTORY]
            [--faildir FAILED_INPUT_DIRECTORY]
            [--poll POLLING_DELAY] [--lock]
            [--log LOG_SPECIFIER] [--loglevel LOG_LEVEL]
            [--verbose] [--version] [--daemon] [--foreground]
            [FILTER_EXPRESSION] [SORT_EXPRESSION]
```

DESCRIPTION

nafilter is the NAF tool suite flow filter and sorter. It acts as a post-processor for flow data aggregated by *nafalize* (1), providing a way to filter and sort aggregated flow data. Most of the operations supported by nafilter can also be performed during aggregation by *nafalize* (1); indeed, nafilter's filter and sort expressions are identical to those used by *nafalize* (1).

Note that unlike *nafalize* (1), nafilter's filter expression applies to aggregated flow values, not raw flow values. This can be used to implement thresholds and other filters on aggregated counts, which cannot be done directly with *nafalize* (1).

As with *nafalize* (1), nafilter's output is in the NAF aggregated flow format, which is an IPFIX message stream containing aggregated flow data. To convert the NAF format to whitespace-delimited text for processing by other tools or for human consumption, use the *nafscii* (1) tool included with the NAF distribution.

nafilter, like all NAF tools, operates by default in **once** mode, though it can also be run as a **daemon**. In daemon mode, nafilter will wait for new input to match its input specifier, and move processed input to the **−−nextdir** directory. This can be used to build ''chains'' of daemons for automated batch processing of flow data.

OPTIONS

Input Options

The input specifier determines where nafilter will read its input from. nafilter defaults to reading from standard input.

−−in *INPUT_SPECIFIER*

INPUT_SPECIFIER is an input specifier. This is a filename, a directory name, a file glob pattern (in which case it should be escaped or quoted to prevent the shell from expanding the glob pattern), or the string **−** to read from standard input.

Output Options

The output specifier determines where nafilter will write its output. The output specifier is optional. If reading standard input, output defaults to standard output. If reading from files on disk, output defaults to one file per input file, named as the input file in the same directory as the input file with a **.naf** extension. Since nafilter reads and writes NAF files (which should both have a **.naf** extension), you should supply at least a directory in **−−out** when reading from multiple files to ensure that the filtered output does not overwrite the input files.

−−out *OUTPUT_SPECIFIER*

OUTPUT_SPECIFIER is an output specifier. If present, this should be a filename or a directory name, or the string **−** to write to standard output.

Daemon Options

These options are used to run nafilter in daemon mode for batch processing of packet and flow files.

−−daemon
Run nafilter in daemon mode. Instead of processing its input then exiting, nafilter will continually look for new input matching its input specifier. This will cause nafilter to fork into the background and exit.

−−foreground
Instead of forking in **−−daemon** mode, stay in the foreground. Useful for debugging.

−−lock
Use lockfiles for concurrent file access protection. Highly recommended in **−−daemon** mode, especially if two NAF daemons are interacting through a given directory.

−−poll *POLLING_DELAY*
POLLING_DELAY is the polling delay in seconds; how long nafilter will wait for new input when none is available. The default is 60 seconds.

−−nextdir *PROCESSED_INPUT_DIRECTORY*
When reading from files, if this option is present, input files will be moved to *PROCESSED_INPUT_DIRECTORY* after they are successfully processed. The special string **delete** will cause successfully processed input to be removed instead. This option is required in daemon mode.

−−faildir *FAILED_INPUT_DIRECTORY*
When reading from files, if this option is present, input files will be moved to *FAILED_INPUT_DIRECTORY* if processing failed. The special string **delete** will cause failed input to be removed instead. This option is required in daemon mode.

Logging Options

These options are used to specify how log messages are routed. nafalize can log to standard error, regular files, or the UNIX syslog facility.

−−log *LOG_SPECIFIER*
Specifies destination for log messages. *LOG_SPECIFIER* can be a *syslog*(3) facility name, the special value **stderr** for standard error, or the *absolute* path to a file for file logging. Standard error logging is only available in **−−daemon** mode if **−−foreground** is present. The default log specifier is **stderr** if available, **user** otherwise.

−−loglevel *LOG_LEVEL*
Specify minimum level for logged messages. In increasing levels of verbosity, the supported log levels are **quiet**, **error**, **critical**, **warning**, **message**, **info**, and **debug**. The default logging level is **warning**.

−−verbose
Equivalent to **−−loglevel debug**.

−−version
If present, print version and copyright information to standard error and exit.

FILTER EXPRESSION SYNTAX

If present, the filter expression determines which aggregated flows will be passed to output. The filter expression defines permitted values for each field in the flow; only if every field in a flow matches the filter does the flow pass the filter. Flows that do not pass the filter are simply dropped.

If no filter expression is given, nafilter simply passes all input to output; this can be used with a sort expression to sort NAF data without filtering it.

The filter expression takes the following form:

filter [**bin** *time-rangelist*] [**sip** [**not**] *address-rangelist*] [**dip** [**not**] *address-rangelist*] [**sp** [**not**] *numeric-rangelist*] [**dp** [**not**] *numeric-rangelist*] [**flows** *numeric-rangelist*] [**rev flows** *numeric-rangelist*] [**packets** *numeric-rangelist*] [**rev packets** *numeric-rangelist*] [**octets** *numeric-rangelist*] [**rev octets** *numeric-rangelist*]

The **bin** rangelist filters flows by bin start time; the bin start time must appear within the rangelist, inclusive, to pass the filter. The **sip**, **dip**, **sp**, **dp**, and **proto** rangelists filter flows by key fields; the **not** keyword on each of these rangelists inverts the rangelist. The **flows**, **packets**, and **octets** and their **rev** (reverse) counterparts filter flows by value fields. These value filters apply to aggregated flow values, not raw flow values, since nafilter can only see aggregated flow data. See **Rangelist Syntax** below for details on rangelists.

Rangelist Syntax

A rangelist is simply a comma-separated list of ranges. Three types of ranges are supported: time, address, and numeric. The supported time ranges (used for **bin** filtering) are as follows:

YYYY−MM−DD
: Matches a single day (UTC).

*YYYY−MM−DD hh***:***mm***:***ss*
: Matches from the given time to the end of the same day (23:59:59 UTC).

YYYY−MM−DD **−** *YYYY−MM−DD*
: Matches from the beginning of the first day through the end of the second (UTC).

*YYYY−MM−DD hh***:***mm***:***ss* **−** *YYYY−MM−DD hh***:***mm***:***ss*
: Matches any time between the first and last times, inclusive.

The supported address ranges (used in **perimeter** mode and for **sip** and **dip** filtering) are as follows:

*a***.***b***.***c***.***d*
: Matches a single IPv4 address.

*a***.***b***.***c***.***d***/***m*
: Matches any IPv4 address in the specified CIDR block.

*a***.***b***.***c***.***d***−***e***.***f***.***g***.***h*
: Matches any IPv4 address between the first and last addresses, inclusive.

The supported numeric ranges are as follows:

n Matches a single integer.

*m***−***n*
: Matches integers between the first and last, inclusive.

<*m* Matches integers less than the given integer.

>*m* Matches integers greater than the given integer.

SORT EXPRESSION SYNTAX

If present, the sort expression defines the sort order of nafilter's output; by default, the output records appear in the same order as the input records. The sort expression takes the following form:

sort (**srcid** | **sip** | **dip** | **proto** | **sp** | **dp** | **octets** | **rev octets** | **packets** | **rev packets** | **flows** | **rev flows** | **src hosts** | **dest hosts** | **src ports** | **dest ports**) [**asc** | **desc**] ... [**limit** *limit-count*]

Multiple fields may appear in the sort expression; subsequent fields are only compared when all previous fields are equal. If **limit** is present, only the first *limit-count* records will be output per time bin. This can be used to build top-N lists.

SIGNALS

nafilter responds to **SIGINT** or **SIGTERM** by terminating input processing, aggregating and flushing any pending sort bins to the current output, and exiting.

BUGS

Known issues are listed in the **README** file in the NAF tools source distribution. Note that NAF should be considered alpha-quality software; not every concievable input and aggregation is exhaustively tested at each release, and specific features may be completely untested. Please be mindful of this before deploying NAF in production environments. Bug reports and feature requests may be sent directly to the author, Brian

Trammell, via email at <bht@cert.org>.

AUTHORS

Brian Trammell <bht@cert.org>, for the CERT Network Situational Awareness Group, http://www.cert.org/netsa.

SEE ALSO

nafalize (1)