NAME

ratop – display and update sorted network flow data

SYNOPSIS

ratop [raoptions] [-- *filter-expression*]

DESCRIPTION

Ratop reads **argus(8)** data from an *argus-file*, or from a remote data source, and periodically displays a sorted list of network flow records. When read from a file, **ratop** displays the resulting flow caches when the file is completed, updating its status display line with each input. When reading from a live argus data stream, **ratop** will display data, asynchronously in realtime, as it is received from the source.

Flow data is aggregated as its read, (see **racluster.1**), resulting in a single line for each network transaction encountered in the data stream. The default sorting key is total packets per flow, but other keys can be used instead. Flow records that have been idle for more than the default 60s are removed. Various output options, such as the specific columns of data to display, the entry idle timeout value, the screen refresh rate, etc ... are all configurable.

ratop uses **ncurses** and **readline.3**, when available, to provide a **vi.1** look and feel for displaying, navigating and modifying network flow data.

While running **ratop** a lot of help can be obtained from the on-line help system, using the ":h" command.

OPTIONS

Command line option specifications are processed from left to right. Options can be specified more than once. If conflicting options are specified, later specifications override earlier ones. This makes it viable to create a shell alias for **ratop** with preferred defaults specified, then override those preferred defaults as desired on the command line.

ratop, like all ra based clients, supports a number of **ra options** including filtering of input argus records through a terminating filter expression, and the ability to specify the output style, format and contents for printing data. See **ra(1)** for a complete description of **ra options**. **ratop(1)** specific options are:

−m *aggregation object*

Supported aggregation objects are:

none	use a null flow key.
srcid	argus source identifier.
smac	source mac(ether) addr.
dmac	destination mac(ether) addr.
soui	oui portion of the source mac(ether) addr.
doui	oui portion of the destination mac(ether) addr.
smpls	source mpls label.
dmpls	destination label addr.
svlan	source vlan label.
dvlan	destination vlan addr.
saddr/[l|m]	source IP addr/[cidr len | m.a.s.k].
daddr/[l|m]	destination IP addr/[cidr len | m.a.s.k].
matrix/l	sorted src and dst IP addr/cidr len.
proto	transaction protocol.
sport	source port number. Implies use of 'proto'.
dport	destination port number. Implies use of 'proto'.
stos	source TOS byte value.
dtos	destination TOS byte value.
sttl	src -> dst TTL value.
dttl	dst -> src TTL value.

stcpb	src -> dst TCP base sequence number.
dtcpb	dst -> src TCP base sequence number.
inode[/l|m]]	intermediate node IP addr/[cidr len | m.a.s.k], source of ICMP mapped events.
sco	source ARIN country code, if present.
dco	destination ARIN country code, if present.
sas	source node origin AS number, if available.
das	destination node origin AS number, if available.
ias	intermediate node origin AS number, if available.

-M *modes*

Supported modes are:

correct	Attempt to correct the direction of flows by also searching the reverse flow key, if a match isn't found in the cache. This mode is on by default when using the default full 5-tuple flow key definitions.
nocorrect	Turn off flow correction for direction. This mode is used by default if the flow key has been changed.
preserve	Preserve fields when aggregating matching flow data.
nopreserve	Do not preserve fields when aggregating matching flow data.
norep	Do not generate an aggregate statistic for each flow. This is used primarily when the output represents a single object. Primarily used when merging status records to generate single flows that represent single transactions.
rmon	Generate data suitable for producing RMON types of metrics.
nocurses	Do not use the curses interface to present data. This option is primarily used when debugging ratop, to get around the issues of screen maniuplation within a debugger like gdb or lldb.

DISPLAY

The first several lines of the **ratop** display show global state. The top line shows how ratop is running, with the list of command line options that are in effect. In the upper most right corner is the current time. The next line is the column title line, that labels each column. The bottom line is the command line, where you will see and prepare ':' commands. The line above the bottom line is the status line, showing the number of flows that are in the **ratop** process queue, display queue, the total number of flows read, the rate of flow records read, and the current status, whether it is Active, reading records, or Idle, when all input is complete. This line can be toggled on or off using ^G.

Flows caches are displayed one per row and are sorted by total pkts, by default. **ratop** sorting can be configured using the *rarc* variable RA_SORT_ALGORITHMS, or by using the ":P" command.

ratop supports 3 basic filters. Like all other ra* programs, **ratop** will send its command line filter to its remote argus data sources, to limit the load on the wire. This is the "remote" filter. Also, **ratop** supports a "local" filter, that is applied to flow record input. Normally this is used when the remote argus data source doesn't support the syntax of the specific filter. **ratop** also support a "display" filter, that is used to select which flow records are to be displayed. This filter does not have any impact on the internal flow caches that **ratop** is tracking, so you can change the "display" filter at any time and see the current state of other flows.

COLOR

ratop supports color which is configured using the rarc file. The RA_COLOR_CONFIG file is a fall through specification of flow filters and field color definitions. For flows that

match a filter, specific fields in the row will be painted the configured color. Because the filter specification supports the " cont " directive, a single row can be painted by any number of color definitions.

When color is enabled **ratop** will attempt to color IP addresses to indicate that local host address, and the local network. This is very helpful in mobile host installations, where you may not know what IP address has been assigned the localhost. **ratop** also supports coloring local addresses based on the RA_LOCAL rarc variable.

See racolor.conf.5.

ARGUS EVENTS

Introduced in argus-3.0.8, **ratop** supports correlating specific ARGUS_EVENT data with flow data, which can be turned on through the use of the RA_CORRELATE_EVENTS rarc variable. **ratop** will process argus-lsof event data generated by host bourne argi, and label flow data with user, pid and process name metadata. While experimental, it is production level functionality, and can be used with other ra* programs to enhance flow data with host os process information. See argus-3.0.8 documentation on ARGUS_EVENTS.

EXAMPLES

ratop -r argus.file -s rank stime dur:14 saddr daddr proto pkts bytes

Read the file argus.file, and display the resulting aggregated and sorted list of flow records, using the default sorting methods.

ratop -S localhost

Run ratop as a live display of realtime flow traffic.

SEE ALSO

rarc(5) racluster(1) racluster.conf(5)