NAME

yaf − Yet Another Flow sensor

SYNOPSIS

```
 yaf     [--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]
         [--live] [--bpf BPF_EXPRESSION] [--rotate-delay ROTATE_DELAY]
         [--idle-timeout IDLE_TIMEOUT]
         [--active-timeout ACTIVE_TIMEOUT]
         [--max-payload PAYLOAD_OCTETS]
         [--max-frags FRAG_TABLE_MAX] [--max-flows FLOW_TABLE_MAX]
         [--uniflow] [--simulate] [--ipfix-tcp | --ipfix-udp]
         [--observation-domain DOMAIN_ID]
```

DESCRIPTION

yaf is Yet Another Flow sensor. It reads packet data from *pcap* (3) dumpfiles as generated by *tcpdump* (1) or via live capture from an interface using *pcap* (3), aggregates these packets into flows, and exports flow records via IPFIX over TCP or UDP, or into serialized IPFIX message streams (IPFIX files) on the local file system.

Since yaf is designed to be deployed on white-box sensors attached to local network segments or span ports at symmetric routing points, it supports bidirectional flow assembly natively. Biflow export is done via a vendor-specific variant of the export method proposed in IETF draft−trammell−ipfix−biflow−02. See the **OUTPUT** section below for information on this format.

yaf also supports partial payload capture, specifically for banner-grabbing applications and protocol verification purposes.

The output of yaf is designed to be collected and manipulated by flow processing toolchains supporting IPFIX. The *yafscii* (1) tool, which is installed with yaf, can also be used to print yaf output in a human-readable format somewhat reminiscient of *tcpdump* (1). yaf output can also be processed and aggregated by the *nafalize* (1) tool, part of NetSA's NAF toolsuite.

OPTIONS

Input Options

The input specifier determines where yaf will read its input from. If reading from a pcap dumpfile (i.e., if **−−live** is not present) and the input specifier is not given, yaf defaults to reading from standard input.

−−in *INPUT_SPECIFIER*

INPUT_SPECIFIER is an input specifier. If reading from a file, 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.

If **−−live** is given, *INPUT_SPECIFIER* is a pcap interface name to capture packets from, and is required.

−−live

If present, capture packets from an interface using libpcap. Requires sufficient privileges for raw packet capture; at this time, yaf is not designed to run setuid or use privilege separation. If present, **−−in** is required, and *INPUT_SPECIFIER* must be a pcap interface name.

−−bpf *BPF_EXPRESSION*

Apply the filter in *BPF_EXPRESSION* to packets as they are read. *BPF_EXPRESSION* must be quoted so the shell will pass it to yaf as a single argument. See the *tcpdump* (1) manpage for information on BPF expressions.

Output Options

The output specifier determines where yaf will send its output. 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 **.yaf** extension. If reading from a live pcap interface, output defaults to files named with the interface name and a timestamp in the current directory.

The **−−ipfix−tcp** and **−−ipfix−udp** options make yaf an IPFIX Exporting Process; in this case, the output specifier is required and specifies the IPFIX Collecting Process to send flow data to.

−−out *OUTPUT_SPECIFIER*
OUTPUT_SPECIFIER is an output specifier. If present, and YAF is not acting as an IPFIX Exporting process, this should be a filename or a directory name, or the string **−** to write to standard output.

If **−−ipfix−udp** or **−−ipfix−tcp** are present, the *OUTPUT_SPECIFIER* specifies the address of the IPFIX Collecting Process to export IPFIX Messages to. This output specifier takes the format *address*[,*port*], where *address* may be an IPv4 address, an IPv6 address, or a hostname that will resolve to an IPv4 or IPv6 address. If *port* is not given, it defaults to port 4739. Note that IPv6 is only available on machines with dual-stack support.

−−rotate−delay *ROTATE_DELAY*
When **−−live** is present, *ROTATE_DELAY* is the rotation delay in seconds between the creation of each subsequent output file. The default delay is 300 seconds (5 minutes).

−−ipfix−udp
If present, operate as an IPFIX Exporting Process sending IPFIX Messages over UDP. Only one of **−−ipfix−udp** or **−−ipfix−tcp** may be present.

−−ipfix−tcp
If present, operate as an IPFIX Exporting Process sending IPFIX Messages over TCP. Only one of **−−ipfix−udp** or **−−ipfix−tcp** may be present.

−−observation−domain *DOMAIN_ID*
Set the observationDomainID on each exported IPFIX message to the given integer value. If not present, the observationDomainId defaults to zero, which, when exporting to IPFIX Collecting Processes, is probably not what you want.

Daemon Options

These options are used to run yaf in daemon mode for batch processing of pcap dumpfiles.

−−daemon
Run yaf in daemon mode. Instead of processing its input then exiting, yaf will continually look for new input matching its input specifier. This will cause yaf to fork into the background and exit. When running **−−live**, the only effect of daemon mode is to cause yaf to fork into the background on startup.

−−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 daemons are interacting through a given directory.

−−poll *POLLING_DELAY*
POLLING_DELAY is the polling delay in seconds; how long yaf 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. yaf 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.

Flow Generation Options

−−idle−timeout *IDLE_TIMEOUT*
Set flow idle timeout in seconds. Flows are considered idle and flushed from the flow table if no packets are recieved for *IDLE_TIMEOUT* seconds. The default flow idle timeout is 300 seconds (5 minutes).

−−active−timeout *ACTIVE_TIMEOUT*
Set flow active timeout in seconds. Any flow lasting longer than *ACTIVE_TIMEOUT* seconds will be flushed from the flow table. The default flow active timeout is 1800 seconds (30 minutes).

−−max−payload *PAYLOAD_OCTETS*
If present, capture at most *PAYLOAD_OCTETS* octets from the start of each direction of each flow. Non-TCP flows will only capture payload from the first packet. If not present, yaf will not attempt to capture payload. If the source pcap data does not have full packet payload, the missing octets will be zero-filled in the output.

−−max−frags *FRAG_TABLE_MAX*
If present, limit the number of outstanding, not-yet reassembled fragments in the fragment table to *FRAG_TABLE_MAX* by prematurely expiring fragments from the table. This option is provided to limit yaf resource usage when operating on data from very large networks or networks with abnormal fragmentation.

−−max−flows *FLOW_TABLE_MAX*
If present, limit the number of open flows in the flow table to *FLOW_TABLE_MAX* by prematurely expiring the flows with the least recently received packets; this is analogous to an adaptive idle timeout. This option is provided to limit YAF resource usage when operating on data from large networks.

−−mac
If present, export MAC-layer information; presently, this only exports VLAN IDs for 802.1q packets. Future revisions of YAF may export MAC addresses, as well.

−−uniflow
If present, export YAF biflows using the Record Adjacency method in section 4.1 of draft−trammell−ipfix−biflow−02. This is useful when exporting to IPFIX Collecting Processes that are not biflow−aware. Given that in the present revision, single-record biflow support in YAF is

vendor−specific, this flag should be given when exporting IPFIX data to any Collecting Process other than *nafalize* (1).

−−simulate
If present, shift dumpfile timestamps to present time and simulate packet interarrival delay. Provided for use in testing, to make a dumpfile appear to be captured live.

OUTPUT

yaf's output consists of an IPFIX message stream. yaf uses a variety of templates for IPFIX data records; the information elements that may appear in these templates are enumerated below:

flowStartMilliseconds IE 152, 8 octets
Flow start time in milliseconds since 1970−01−01 00:00:00 UTC. Always present.

flowEndMilliseconds IE 153, 8 octets
Flow end time in milliseconds since 1970−01−01 00:00:00 UTC. Always present.

octetTotalCount IE 85, 8 octets
Number of octets in packets in forward direction of flow. Always present. May be encoded in 4 octets using IPFIX reduced-length encoding.

reverseOctetTotalCount CERT (PEN 6871) IE 12, 8 octets
Number of octets in packets in reverse direction of flow. Present if flow has a reverse direction. May be encoded in 4 octets using IPFIX reduced-length encoding.

packetTotalCount IE 86, 8 octets
Number of packets in forward direction of flow. Always present. May be encoded in 4 octets using IPFIX reduced-length encoding.

reversePacketTotalCount CERT (PEN 6871) IE 13, 8 octets
Number of packets in reverse direction of flow. Present if flow has a reverse direction. May be encoded in 4 octets using IPFIX reduced-length encoding.

reverseFlowDeltaMilliseconds CERT (PEN 6871) IE 21, 4 octets
Difference in time in milliseconds between first packet in forward direction and first packet in reverse direction. Correlates with (but does not necessarily represent) round-trip time. Present if flow has a reverse direction.

sourceIPv4Address IE 8, 4 octets
IP address of flow source or biflow initiator. Always present.

destinationIPv4Address IE 12, 4 octets
IP address of flow source or biflow responder. Always present.

sourceTransportPort IE 7, 2 octets
TCP or UDP port on the flow source or biflow initiator endpoint. Always present.

destinationTransportPort IE 11, 2 octets
TCP or UDP port on the flow destination or biflow responder endpoint. Always present. For ICMP flows, contains ICMP type * 256 + ICMP code. This is non−standard, and an open issue in yaf.

protocolIdentifier IE 4, 1 octet
IP protocol of the flow. Always present.

flowEndReason IE 136, 1 octet
Flow end reason code, as defined by the IPFIX Information Model. Always present.

tcpSequenceNumber IE 184, 4 octets
Initial sequence number of the forward direction of the flow. Present if the flow's protocolIdentifier is 6 (TCP).

reverseTcpSequenceNumber CERT (PEN 6871) IE 20, 4 octets
Initial sequence number of the reverse direction of the flow. Present if the flow's protocolIdentifier is 6 (TCP) and the flow has a reverse direction.

initialTCPFlags CERT (PEN 6871) IE 14, 1 octet
: TCP flags of initial packet in the forward direction of the flow. Present if the flow's protocolIdentifier is 6 (TCP).

unionTCPFlags CERT (PEN 6871) IE 15, 1 octet
: Union of TCP flags of all packets other than the initial packet in the forward direction of the flow. Present if the flow's protocolIdentifier is 6 (TCP).

reverseInitialTCPFlags CERT (PEN 6871) IE 16, 1 octet
: TCP flags of initial packet in the reverse direction of the flow. Present if the flow's protocolIdentifier is 6 (TCP) and the flow has a reverse direction.

reverseUnionTCPFlags CERT (PEN 6871) IE 17, 1 octet
: Union of TCP flags of all packets other than the initial packet in the reverse direction of the flow. Present if the flow's protocolIdentifier is 6 (TCP) and the flow has a reverse direction.

vlanId IE 58, 2 octets
: 802.1q VLAN tag of the first packet in the forward direction of the flow. Present if MAC layer export is enabled and a VLAN tag is present.

reverseVlanId CERT (PEN 6871) IE 27, 2 octets
: 802.1q VLAN tag of the first packet in the reverse direction of the flow. Present if MAC layer export is enabled, the flow has a reverse direction, and a VLAN tag is present.

payload CERT (PEN 6871) IE 18, variable-length
: Initial *n* bytes of forward direction of flow payload. Present if payload collection is enabled and payload is present in the forward direction of the flow.

reversePayload CERT (PEN 6871) IE 19, variable-length
: Initial *n* bytes of reverse direction of flow payload. Present if payload collection is enabled and payload is present in the reverse direction of the flow.

SIGNALS

yaf responds to **SIGINT** or **SIGTERM** by terminating input processing, flushing any pending flows to the current output, and exiting.

BUGS

Known issues are listed in the **README** file in the YAF tools source distribution. Note that YAF should be considered alpha-quality software; not every concievable input and optionis exhaustively tested at each release, and specific features may be completely untested. Please be mindful of this before deploying YAF in production environments. YAF's output format may also change, as the development of YAF is intended to track progress in the IPFIX working group; the file output of YAF should not presently be used for archival storage of flow data. 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

yafscii (1), *tcpdump* (1), *pcap* (3), *nafalize* (1), and the following IETF Internet−Drafts: draft−ietf−ipfix−protocol, draft−ietf−ipfix−info, draft−trammell−ipfix−biflow.