
Tranalyzer2

txtSink



Text Output



Tranalyzer Development Team

Contents

1	txtSink	1
1.1	Description	1
1.2	Dependencies	1
1.3	Configuration Flags	1
1.4	Additional Output	2

1 txtSink

1.1 Description

The txtSink plugin provides human readable text output which can be saved in a file `PREFIX_flows.txt`, where `PREFIX` is provided via the `-w` option. The generated output contains a textual representation of all plugins results. Each line in the file represents one flow. The different output statistics of the plugins are separated by a tab character to provide better post-processing with command line scripts or statistical toolsets.

1.2 Dependencies

1.2.1 External Libraries

If gzip compression is activated (`TFS_GZ_COMPRESS=1`), then **zlib** must be installed.

TFS_GZ_COMPRESS=1		
Ubuntu:	<code>sudo apt-get install</code>	<code>zlib1g-dev</code>
Arch:	<code>sudo pacman -S</code>	<code>zlib</code>
Gentoo:	<code>sudo emerge</code>	<code>zlib</code>
openSUSE:	<code>sudo zypper install</code>	<code>zlib-devel</code>
Red Hat/Fedora¹:	<code>sudo dnf install</code>	<code>zlib-devel</code>
macOS²:	<code>brew install</code>	<code>zlib</code>

1.2.2 Core Configuration

This plugin requires the following core configuration:

- `$T2HOME/tranalyzer2/src/tranalyzer.h:`
 - `BLOCK_BUF=0`

1.3 Configuration Flags

The configuration flags for the txtSink plugins are separated in two files.

1.3.1 txtSink.h

Name	Default	Description
<code>TFS_SPLIT</code>	1	Split the output file (Tranalyzer <code>-W</code> option)
<code>TFS_PRI_HDR</code>	1	Print a row with column names at the start of the flow file
<code>TFS_HDR_FILE</code>	1	Generate a separate header file (Section 1.4.1)
<code>TFS_PRI_HDR_FW</code>	0	Print header in every output fragment (Tranalyzer <code>-W</code> option)
<code>TFS_GZ_COMPRESS</code>	0	Compress the output (gzip)
<code>TFS_FLOWS_TXT_SUFFIX</code>	<code>"_flows.txt"</code>	Suffix for the flow file

¹If the `dnf` command could not be found, try with `yum` instead

²Brew is a packet manager for macOS that can be found here: <https://brew.sh>

Name	Default	Description
TFS_HEADER_SUFFIX	"_headers.txt"	Suffix for the header file

1.3.2 bin2txt.h

`bin2txt.h` controls the conversion from internal binary format to standard text output.

Name	Default	Description
IP4_FORMAT	0	IPv4 addresses representation: 0: normal, 1: normalized (padded with zeros), 2: one 32-bits hex number 3: one 32-bits unsigned number
IP6_FORMAT	0	IPv6 addresses representation: 0: compressed, 1: uncompressed, 2: one 128-bits hex number, 3: two 64-bits hex numbers
MAC_FORMAT	0	MAC addresses representation: 0: normal (edit <code>MAC_SEP</code> to change the separator), 1: one 64-bits hex number,
MAC_SEP	": "	Separator to use in MAC addresses: 11:22:33:44:55:66
B2T_NON_IP_STR	"-"	Representation of non-IPv4/IPv6 addresses in IP columns
HEX_CAPITAL	0	Hex output: 0: lower case; 1: upper case
TFS_EXTENDED_HEADER	0	Extended header in flow file
B2T_NANOSECS	0	Time precision: 0: microsecs, 1: nanosecs
TFS_NC_TYPE	2	Types in header file: 0: none, 1: numbers, 2: C types
TFS_SAN_UTF8	1	Activates the UTF-8 sanitizer for strings
B2T_TIMESTAMP	0	Print unix timestamps as human readable dates
HDR_CHR	"%"	start character(s) of comments
SEP_CHR	"\t"	column separator in the flow file ";", ".", "_", and "\" should not be used

1.3.3 Environment Variable Configuration Flags

The following configuration flags can also be configured with environment variables (`ENVCTRL>0`):

- `TFS_FLOWS_TXT_SUFFIX`
- `TFS_HEADER_SUFFIX`

1.4 Additional Output

1.4.1 Header File

The header file `PREFIX_headers.txt` describes the columns of the flow file and provides some additional information, such as plugins loaded and PCAP file or interface used, as depicted below. The default suffix used for the header file is `_headers.txt`. This suffix can be configured using `TFS_HEADER_SUFFIX`.

```

# Date: 1566316839.259591 sec (Tue 5 Aug 2023 18:00:39 CEST)
# Tranalyzer 0.9.0 (Anteater), Cobra.
# Core configuration: L2, IPv4, IPv6
# SensorID: 666
# PID: 13221
# Command line: /home/user/tranalyzer2-0.9.0/tranalyzer2/src/tranalyzer -r file.pcap
# HW Info: hostname;sysname;release;version;machine
# SW info: libpcap version 1.9.1
#
# Plugins loaded:
# 01: protoStats, version 0.9.0
# 02: basicFlow, version 0.9.0
# 03: macRecorder, version 0.9.0
# 04: portClassifier, version 0.9.0
# 05: basicStats, version 0.9.0
# 06: tcpFlags, version 0.9.0
# 07: tcpStates, version 0.9.0
# 08: icmpDecode, version 0.9.0
# 09: connStat, version 0.9.0
# 10: txtSink, version 0.9.0
#
# Col No.   Type           Name                Description
1           C              dir                 Flow direction
2           U64          flowInd             Flow index
3           H64          flowStat            Flow status and warnings
4           U64.U32       timeFirst           Date time of first packet
5           U64.U32       timeLast            Date time of last packet
6           U64.U32       duration            Flow duration
7           U8           numHdrDesc          Number of different headers descriptions
8           U16:R        numHdrs             Number of headers (depth) in hdrDesc
9           SC:R        hdrDesc             Headers description
10          MAC:R        srcMac              Mac source
11          MAC:R        dstMac              Mac destination
12          H16          ethType             Ethernet type
13          U16:R        vlanID              VLAN IDs
14          IPX         srcIP               Source IP address
15          SC           srcIPCC             Source IP country
16          S           srcIPOrg             Source IP organization
17          U16          srcPort             Source port
18          IPX         dstIP               Destination IP address
19          SC           dstIPCC             Destination IP country
20          S           dstIPOrg            Destination IP organization
21          U16          dstPort             Destination port
22          U8           l4Proto             Layer 4 protocol
23          H8           macStat             macRecorder status
...

```

The column number can be used, e.g., with `awk` or `tawk` to query a given column. For example, to extract all ICMP flows (layer 4 protocol equals 1) from a flow file:

```
awk -F'\t' '$22 == 1' PREFIX_flows.txt
```

The second column indicates the type of the column (see table below). If the value is repetitive, the type is postfixed with `:R`. Repetitive values can occur any number of times (from 0 to N). Each repetition is separated by a semicolon. The `_'` indicates a compound, i.e., a value containing 2 or more subvalues.

#	Name	Description	#	Name	Description	#	Name	Description
1	I8	int8	11	U128	uint128	21	LD	long double
2	I16	int16	12	U256	uint256	22	C	char
3	I32	int32	13	H8	hex8	23	S	string
4	I64	int64	14	H16	hex16	24	C	flow direction ³
5	I128	int128	15	H32	hex32	25	TS	timestamp ⁴
6	I256	int256	16	H64	hex64	26	U64.U32	duration
7	U8	uint8	17	H128	hex128	27	MAC	mac address
8	U16	uint16	18	H256	hex256	29	IP4	IPv4 address
9	U32	uint32	19	F	float	29	IP6	IPv6 address
10	U64	uint64	20	D	double	30	IPX	IPv4 or 6 address
						31	SC	string class ⁵

³A: client→server, B: server→client

⁴U64.U32/S (See B2T_TIMESTR in bin2txt.h)

⁵string without quotes