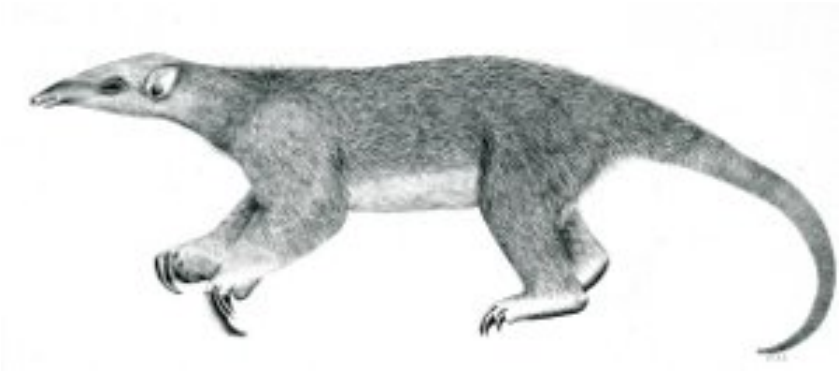
Tranalyzer2

binSink

Binary Output



Tranalyzer Development Team

Contents

1 binSink

1.1 Description

The binSink plugin is one of the basic output plugin for Tranalyzer2. It uses the output prefix (`-w` option) to generate a binary flow file with suffix `_flows.bin`. All standard output from every plugin is stored in binary format in this file.

1.2 Dependencies

1.2.1 External Libraries

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

Kali/Ubuntu: `sudo apt-get install zlib1g-dev`

Arch: `sudo pacman -S zlib`

Fedora/Red Hat: `sudo yum install zlib-devel`

Gentoo: `sudo emerge zlib`

OpenSUSE: `sudo zypper install zlib-devel`

Mac OS X: `brew install zlib`[1]

1.3 Configuration Flags

The following flags can be used to control the output of the plugin:

Name	Default	Description
GZ_COMPRESS	0	Compress the output (gzip)
SFS_SPLIT	1	Split the output file (Tranalyzer `-W` option)
FLOWS_SUFFIX	"_flows.bin"	Suffix to use for the output file
STD_BUFSHFT	BUF_DATA_SHFT * 4	

1.4 Post-Processing

1.4.1 tranalyzer-b2t

The program `tranalyzer-b2t` can be used to transform binary Tranalyzer files into text or json files. The converted file uses the same format as the one generated by the `txtSink` or `jsonSink` plugin.

The program can be found in `$T2HOME/utils/tranalyzer-b2t/` and can be compiled by typing `make`.

[1] Brew is a packet manager for Mac OS X that can be found here: https://brew.sh

The use of the program is straightforward:

- bin→txt: `./tranalyzer-b2t -r FILE_flows.bin -w FILE_flows.txt`
- bin→json: `./tranalyzer-b2t -r FILE_flows.bin -j -w FILE_flows.json`

If the `-w` option is omitted, the destination default to stdout.
Additionally, the `-n` option can be used **not** to print the name of the columns as the first row.

1.5 Custom File Output

- `PREFIX_flows.bin`: Binary representation of Tranalyzer output