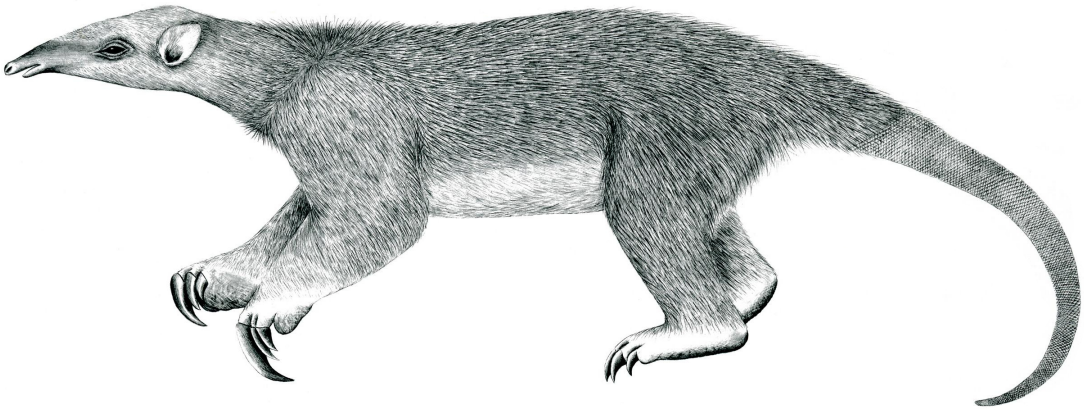
Tranalyzer2

Getting Started

Installation



Tranalyzer Development Team

Contents

1 Introduction **1**
1.1 Getting Tranalyzer 1
1.2 Dependencies 1
1.3 Compilation 2
1.4 Installation 3
1.5 Getting Started 3
1.6 Getting Help 3

1 Introduction

Tranalyzer2 is a lightweight flow generator and packet analyzer designed for simplicity, performance and scalability. The program is written in C and built upon the *libpcap* library. It provides functionality to pre- and post-process IPv4/IPv6 data into flows and enables a trained user to see anomalies and network defects even in very large datasets. It supports analysis with special bit coded fields and generates statistics from key parameters of IPv4/IPv6 tcpdump traces either being live-captured from an Ethernet interface or one or several pcap files. The quantity of binary and text based output of Tranalyzer2 depends on enabled modules, herein denoted as **plugins**. Hence, users have the possibility to tailor the output according to their needs and developers can develop additional plugins independent of the functionality of other plugins.

1.1 Getting Tranalyzer

Tranalyzer can be downloaded from: https://tranalyzer.com/downloads.html

1.2 Dependencies

Tranalyzer2 requires the following tools and libraries:

Kali/Ubuntu:

```
sudo apt-get install autoconf autoconf-archive automake libbsd-dev libpcap-dev
    libreadline-dev libtool make meson zlib1g-dev
```

Arch/Manjaro:

```
sudo pacman -S autoconf autoconf-archive automake bash-completion gcc libpcap
    libtool make meson pkgconf zlib
```

CentOS/Fedora/Red Hat:

```
sudo dnf install autoconf autoconf-archive automake bzip2 libbsd-devel
    libpcap-devel libtool meson readline-devel zlib-devel [1]
```

Gentoo:

```
sudo emerge autoconf autoconf-archive automake bash-completion libpcap libtool meson zlib
```

openSUSE:

```
sudo zypper install autoconf autoconf-archive automake gcc libbsd-devel
    libpcap-devel libtool meson readline-devel zlib-devel
```

macOS:

```
brew install autoconf autoconf-archive automake libpcap libtool meson readline zlib [2]
```

Note that `meson` is optional, but recommended as it is much faster than the autotools (`autoconf`, `automake`, ...).

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

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

1.3 Compilation

To build Tranalyzer2 and the plugins, run one of the following commands:

- Tranalyzer2 only:
  `cd "$T2HOME"; ./autogen.sh tranalyzer2`
  (alternative: `cd "$T2HOME/tranalyzer2"; ./autogen.sh`)
- A specific plugin only, e.g., `myPlugin`:
  `cd "$T2HOME"; ./autogen.sh myPlugin`
  (alternative 1: `cd "$T2PLHOME/myPlugin"; ./autogen.sh`)
  (alternative 2: `cd "$T2HOME/plugins/myPlugin"; ./autogen.sh`)
- Tranalyzer2 and a default set of plugins:
  `cd "$T2HOME"; ./autogen.sh`
- Tranalyzer2 and all the plugins in `T2HOME`:
  `cd "$T2HOME"; ./autogen.sh -a`
- Tranalyzer2 and a custom set of plugins (listed in plugins.build) (Section 1.3.1):
  `cd "$T2HOME"; ./autogen.sh -b`

where `T2HOME` points to the root folder of Tranalyzer, i.e., where the file `README.md` is located.

For finer control of which plugins to load, refer to Tranalyzer2 documentation .

Note that if `t2_aliases` is installed, the `t2build` command can be used instead of `autogen.sh`. The command can be run from anywhere, so just replace the above commands with `t2build tranalyzer2`, `t2build myPlugin`, `t2build -a` and `t2build -b`. Run `t2build --help` for the full list of options accepted by the script.

1.3.1 Custom Build

The `-b` option of the `autogen.sh` script takes an optional file name as argument. If none is provided, then the default `plugins.build` is used. The format of the file is as follows:

- Empty lines and lines starting with a `'#'` are ignored (can be used to prevent a plugin from being built)
- One plugin name per row
- Example:

```
# Do not build the tcpStates plugin
#tcpStates

# Build the txtSink plugin
txtSink
```

A `plugins.ignore` file can also be used to prevent specific plugins from being built. A different filename can be used with the `-I` option.

1.4 Installation

The `-i` option of the `autogen.sh` script installs Tranalyzer in `/usr/local/bin` (as `tranalyzer`) and the man page in `/usr/local/man/man1`. Note that root rights are required for the installation.

Alternatively, use the file `t2_aliases` or add the following alias to your `~/.bash_aliases`:

```
alias tranalyzer="$T2HOME/tranalyzer2/src/tranalyzer"
```

where `T2HOME` points to the root folder of Tranalyzer, i.e., where the file `README.md` is located.

The man page can also be installed manually, by calling (as root):

```
mkdir -p /usr/local/man/man1 && gzip -c man/tranalyzer.1 > /usr/local/man/man1/tranalyzer.1.gz
```

1.4.1 Aliases

The file `t2_aliases` documented in `$T2HOME/scripts/doc/scripts.pdf` contains a set of aliases and functions to facilitate working with Tranalyzer. To install it, append the following code to `~/.bashrc` or `~/.bash_aliases` (make sure to replace `$T2HOME` with the actual path, e.g., `$HOME/tranalyzer2-0.9.3`):

```
if [ -f "$T2HOME/scripts/t2_aliases" ]; then
    . "$T2HOME/scripts/t2_aliases"            # Note the leading `.'
fi
```

1.5 Getting Started

Run Tranalyzer as follows:

```
tranalyzer -r file.pcap -w outfolder/outprefix
```

For a full list of options, use Tranalyzer `-h` option: `tranalyzer -h` or refer to the complete documentation.

1.6 Getting Help

1.6.1 Documentation

Tranalyzer and every plugin come with their own documentation, which can be found in the `doc` subfolder. The complete documentation of Tranalyzer2 and all the locally available plugins can be generated by running `make` in `$T2HOME/doc`. The file `t2_aliases` provides the function `t2doc` to allow easy access to the different parts of the documentation from anywhere.

1.6.2 Man Page

If the man page was installed (Section 1.4), then accessing the man page is as simple as calling

```
man tranalyzer
```

If it was not installed, then the man page can be invoked by calling

```
man $T2HOME/tranalyzer2/man/tranalyzer.1
```

1.6.3 Help

For a full list of options, use Tranalyzer `-h` option: `tranalyzer -h`

1.6.4 FAQ

Refer to the complete documentation in `$T2HOME/doc` for a list of frequently asked questions.

1.6.5 Contact

Any feedback, feature requests and questions are welcome and can be sent to the development team via email at:

andy@tranalyzer.com