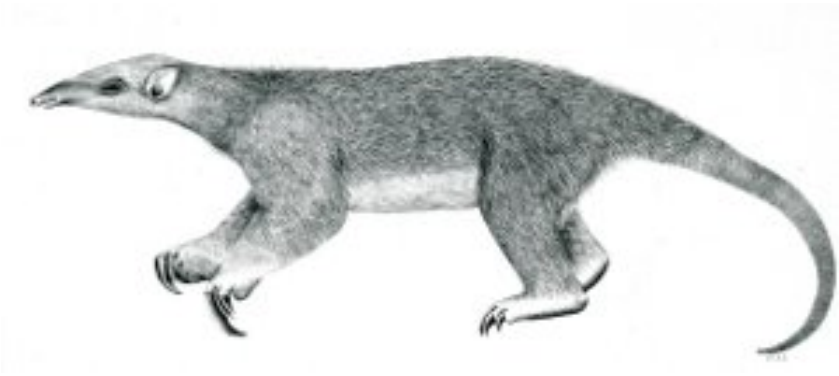
Tranalyzer2

PDF Report Generation from PCAP using t2fm

Tutorial



Tranalyzer Development Team

Contents

1 PDF Report Generation from PCAP using t2fm

1.1 Introduction

This tutorial presents `t2fm`, a script which generates a PDF report out of a PCAP file. Information provided in the report includes top source and destination addresses and ports, protocols and applications, DNS and HTTP activity and potential warnings, such as executable downloads or SSH connections.

1.2 Prerequisites

For this tutorial, it is assumed the user has a basic knowledge of Tranalyzer and that the file `t2_aliases` has been sourced in `~/.bashrc` or `~/.bash_aliases` as follows[1] (make sure to replace `$T2HOME` with the actual path, e.g., `$HOME/tranalyzer2-0.7.0lm1/trunk`):

```
# $HOME/.bashrc

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

1.2.1 Required plugins

The following plugins must be loaded for `t2fm` to produce a useful report:

- `basicFlow`
- `basicStats`
- `txtSink`

1.2.2 Optional plugins

The following plugins are optional:

- `arpDecode`
- `dnsDecode`
- `geoip`
- `pwX`
- `sshDecode`
- `sslDecode`
- `httpSniffer` , configured as follows[2]:
  - `HTTP_SAVE_IMAGE=1`
  - `HTTP_SAVE_VIDEO=1`
  - `HTTP_SAVE_AUDIO=1`
  - `HTTP_SAVE_MSG=1`
  - `HTTP_SAVE_TEXT=1`
  - `HTTP_SAVE_APPL=1`
- `nDPI` , configured as follows:
  - `NDPI_OUTPUT_STR=1`
- `portClassifier` , configured as follows:
  - `PBC_NUM=1`
  - `PBC_STR=1`

If one of those plugin is not loaded, messages like `N/A: dnsDecode plugin required` will be displayed in the PDF where the information could not be accessed.

[1] Refer to the file `README.md` or to the documentation for more details
[2] This is only required to report information about EXE downloaded

1.2.3 Packages

The following packages are required to build the PDF:

- texlive-latex-extra
- texlive-fonts-recommended

1.3 Step-by-Step Instructions (PCAP to PDF)

For simplicity, this tutorial assumes the user wants a complete report, i.e., requires all of the optional plugins.

1. Make sure all the plugins are configured as described in Section 1.2
2. Build Tranalyzer and the plugins [3]:
   `t2build tranalyzer2 basicFlow basicStats txtSink arpDecode dnsDecode geoip \`
   `httpSniffer nDPI portClassifier pwX sshDecode sslDecode`
   (Note that those first two steps can be omitted if `t2fm -b` option is used)
3. Run `t2fm` directly on the PCAP file (the report will be named `file.pdf`):
   `t2fm -r file.pcap`
4. Open the generated PDF report `file.pdf`:
   `evince file.pdf`

1.4 Step-by-Step Instructions (flow file to PDF)

Alternatively, if you prefer to run Tranalyzer yourself or already have access to a flow file, replace step 3 with the following steps:

1. Follow point 1 and 2 from Section 1.3
2. Run Tranalyzer on a pcap file as follows:
   `t2 -r file.pcap -w out`
3. The previous command should have created the following files:
   `out_headers.txt`
   `out_flows.txt`
4. Run the `t2fm` script on the flow file generated previously:
   `t2fm -F out_flows.txt`

1.5 Step-by-Step Instructions (MongoDB / PostgreSQL to PDF)

If the `mongoSink` or `psqlSink` plugins were loaded, t2fm can use the created databases to generate the report (faster).

1. Follow point 1 and 2 from Section 1.3[4]
2. Build the `mongoSink` or `psqlSink` plugin:
   - **mongoDB:** `t2build mongoSink`

[3] Hint: use the tab completion to avoid typing the full name of all the plugins: `t2build tr<tab> ... ht<tab> ...`
[4] `HTTP_SAVE_*` do not need to be set as EXE downloads detection is currently not implemented in the DB backends

- **postgreSQL:** `t2build psqlSink`

3. Run Tranalyzer on a pcap file as follows:
   `t2 -r file.pcap -w out`
4. Run the `t2fm` script on the database generated previously:
   - **mongoDB:** `t2fm -m tranalyzer`
   - **postgreSQL:** `t2fm -p tranalyzer`

When generating a report from a database a time range to query can be specified with the `-T` option. The complete format is as follows: `YYYY-MM-DD HH:MM:SS.USEC([+-]OFFSET|Z)`, e.g., `2018-10-01 12:34:56.912345+0100`. Note that only the required fields must be specified, e.g., `2018-09-01` is equivalent to `2018-09-01 00:00:00.000000`. For example, to generate a report from the 1st of September to the 11. of October 2018 at 14:59 from a PostgreSQL database, run the following command: `t2fm -p tranalyzer -T "2018-09-01" "2018-10-11 14:59"`

1.6 Conclusion

This tutorial has presented how `t2fm` can be used to create a PDF report summarising the traffic contained in a PCAP file. Although not discussed in this tutorial, it is also possible to use `t2fm` on a live interface (`-i` option) or on a list of PCAP files (`-R` option). For more details, refer to `t2fm` man page or use `t2fm --help`.