

---

# Tranalyzer2

Generate PDF report out of PCAP files or NetFlow records



Tutorial

---



Tranalyzer Development Team

Contents

1 PDF Report Generation from PCAP using t2fm 1

1.1 Introduction . . . . . 1

1.2 Prerequisites . . . . . 1

1.3 PCAP to PDF in One Command . . . . . 2

1.4 Step-by-Step Instructions (PCAP to PDF) . . . . . 2

1.5 Step-by-Step Instructions (flow file to PDF) . . . . . 2

1.6 Step-by-Step Instructions (ClickHouse / MongoDB / PostgreSQL to PDF) . . . . . 3

1.7 NetFlow Records to PDF in One Command . . . . . 3

1.8 Conclusion . . . . . 3

# 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<sup>1</sup> (make sure to replace `$T2HOME` with the actual path, e.g., `$HOME/tranalyzer2-0.9.3`):

```
# $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:

- |                          |                                                                   |                                                        |
|--------------------------|-------------------------------------------------------------------|--------------------------------------------------------|
| • <code>arpDecode</code> | • <code>httpSniffer</code> , configured as follows <sup>2</sup> : | • <code>nDPI</code> , configured as follows:           |
| • <code>dnsDecode</code> | – <code>HTTP_SAVE_IMAGE=1</code>                                  | – <code>NDPI_OUTPUT_STR=1</code>                       |
| • <code>geoip</code>     | – <code>HTTP_SAVE_VIDEO=1</code>                                  |                                                        |
| • <code>pwX</code>       | – <code>HTTP_SAVE_AUDIO=1</code>                                  | • <code>portClassifier</code> , configured as follows: |
| • <code>sshDecode</code> | – <code>HTTP_SAVE_MSG=1</code>                                    | – <code>PBC_NUM=1</code>                               |
| • <code>sslDecode</code> | – <code>HTTP_SAVE_TEXT=1</code>                                   | – <code>PBC_STR=1</code>                               |
|                          | – <code>HTTP_SAVE_APPL=1</code>                                   |                                                        |

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.

---

<sup>1</sup>Refer to the file `README.md` or to the documentation for more details

<sup>2</sup>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 PCAP to PDF in One Command

For simplicity, this tutorial assumes the user wants a complete report, i.e., requires all of the optional plugins. The `-b` option builds and configures Tranalyzer and the plugins, the `-A` option opens the generated report.

1. Run `t2fm` directly on the PCAP file (the report will be named `file.pdf`):

```
t2fm -b -A -r file.pcap
```

## 1.4 Step-by-Step Instructions (PCAP to PDF)

Alternatively if you prefer to configure Tranalyzer, build the plugins and open the generated report yourself:

1. Make sure all the plugins are configured as described in [Section 1.2](#)
2. Build Tranalyzer and the plugins <sup>3</sup>:  

```
t2build tranalyzer2 basicFlow basicStats txtSink arpDecode dnsDecode geoup \
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` (Note that this step can be omitted if `t2fm -A` option is used):  

```
evince file.pdf
```

## 1.5 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 of [Section 1.4](#) with the following steps:

1. Run Tranalyzer on a pcap file as follows:  

```
t2 -r file.pcap -w out
```
2. The previous command should have created the following files:  

```
out_headers.txt
out_flows.txt
```
3. Run the `t2fm` script on the flow file generated previously:  

```
t2fm -F out_flows.txt
```

---

<sup>3</sup>Hint: use the tab completion to avoid typing the full name of all the plugins: `t2build tr<tab> ... ht<tab> ...`

## 1.6 Step-by-Step Instructions (ClickHouse / MongoDB / PostgreSQL to PDF)

If the `clickhouseSink`, `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 1.4<sup>4</sup>
2. Build the `clickhouseSink`, `mongoSink` or `psqlSink` plugin:

- **ClickHouse:** `t2build clickhouseSink`
- **MongoDB:** `t2build mongoSink`
- **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:

- **ClickHouse:** `t2fm -C tranalyzer`
- **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., `2022-08-23 12:34:56.912345+0100`. Note that only the required fields must be specified, e.g., `2022-08-23` is equivalent to `2022-08-23 00:00:00.000000`. For example, to generate a report from the 1st of July to the 11. of August 2022 at 14:59 from a PostgreSQL database, run the following command: `t2fm -p tranalyzer -T "2022-07-01" "2022-08-11 14:59"`

## 1.7 NetFlow Records to PDF in One Command

For simplicity, this tutorial assumes the user wants a complete report. The `-A` option opens the generated report.

1. Run `t2fm` directly on the NetFlow records (file or directory) (the report will be named `2024.pdf`):

```
t2fm -A -N netflow/2024
```

## 1.8 Conclusion

This tutorial has presented how `t2fm` can be used to create a PDF report summarizing the traffic contained in a PCAP file or NetFlow records. 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`.

---

<sup>4</sup>`HTTP_SAVE_*` do not need to be set as EXE downloads detection is currently not implemented in the DB backends