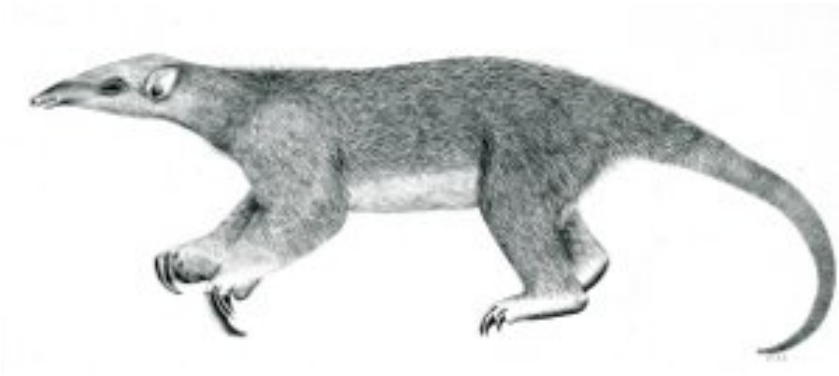
Tranalyzer2

pcapd

Creates PCAP Files



Tranalyzer Development Team

Contents

1 pcapd

1.1 Description

The pcapd plugin can be used to create PCAP files based on some criteria such as flow indexes (Section 1.3.1) or alarms raised by other plugins (Section Section 1.3.2).

1.2 Dependencies

If `PD_MODE=4`, the libpcap version must be at least 1.7.2. (In this mode, the plugin uses the `pcap_dump_open_append()` function which was introduced in the libpcap in February 12, 2015.)

1.3 Configuration Flags

The following flags can be used to configure the plugin:

Variable	Default	Description	Flags
`PD_MODE_IN`	0	0: extract flows listed in input file (if `-e` option was used), or extract flows if alarm bit is set (if `-e` option was not used) 1: dump all packets	
`PD_EQ`	1	whether to save matching (1) or non-matching (0) flows	`PD_MODE_IN=0`
`PD_MODE_OUT`	0	0: one pcap 1: one pcap per flow	
`PD_SPLIT`	1	Split the output file (Tranalyzer `-W` option)	
`PD_FORMAT`	0	Format of the input file (`-e` option): 0: flow index only, 1: flow file format	
`PD_MAX_FD`	128	Maximum number of simultaneously open file descriptors	`PD_MODE_OUT=1`
`PD_SUFFIX`	"`.pcap`"	pcap file extension	

1.3.1 PD_MODE_IN=0, –e option used

The idea behind this mode (`PD_MODE_IN=0` and Tranalyzer `-e` option used) is to use `awk` to extract flows of interest and then the pcapd plugin to create one or more PCAP with all those flows. The format of the file must be as follows:

`PD_FORMAT=0`	The first column must be the flow index (the rest (optionnal) is ignored): `1234 ...`
`PD_FORMAT=1`	The second column must be the flow index: `A 1234 ...`

Lines starting with `'%'`, `'#'`, a space or a tab are ignored, along with empty lines.

Flows whose index appears in the `-e` file will be dumped in a file named `PREFIX_PD_SUFFIX`, where `PREFIX` is the value given to Tranalyzer `-e` option. Note that if `PD_EQ=0`, then flows whose index does **not** appear in the file will be dumped.

1.3.2 PD_MODE_IN=0, –e option not used

In this mode (`PD_MODE_IN=0` and Tranalyzer `-e` option **NOT** used), every flow whose status bit `FL_ALARM=0x20000000` is set (`PD_EQ=1`) or not set (`PD_EQ=0`) will be dumped in a file named `PREFIX_PD_SUFFIX`, where `PREFIX` is the value given to Tranalyzer `-w` or `-W` option.

1.3.3 PD_MODE_IN=1

In this mode, all the packets are dumped into one or more PCAP files. If Tranalyzer `-W` option is used, then the pcap files will be split accordingly. For example, the following command will create PCAP files of 100MB each: `tranalyzer -i eth0 -W out:100M`

1.3.4 PD_MODE_OUT=1

In this mode, every flow will have its own PCAP file, whose name will end with the flow index.

1.4 Additional Output

A PCAP file with suffix `PD_SUFFIX` will be created. The prefix and location of the file depends on the configuration of the plugin.

- If Tranalyzer `-e` option was used, the file is named according to the `-e` option.
- Otherwise the file is named according to the `-w` or `-W` option.

1.5 Examples

For the following examples, it is assumed that Tranalyzer was run as follows, with the *basicFlow* and *txtSink* plugins in their default configuration:

```
tranalyzer -r file.pcap -w out
```

The column numbers can be obtained by looking in the file `out_headers.txt` or by using `tawk` .

1.5.1 Extracting ICMP Flows

To create a PCAP file containing ICMP flows only, proceed as follows:

1. Identify the *"Layer 4 protocol"* column in `out_headers.txt` (column 14):
   `grep "Layer 4 protocol" out_headers.txt`
2. Extract all flow indexes whose protocol is ICMP (1):
   `awk -F'\t' '$14 == 1 { print $2 }' out_flows.txt > out_icmp.txt`
3. Configure pcapd.h as follows: `PD_MODE_IN=0, PD_EQ=1`
4. Build the pcapd plugin: `cd $T2HOME/pcapd/; ./autogen.sh`
5. Re-run Tranalyzer with the `-e` option:
   `tranalyer -r file.pcap -w out -e out_icmp.txt`
6. The file `out_icmp.txt.pcap` now contains all the ICMP flows.

1.5.2 Extracting Non-ICMP Flows

To create a PCAP file containing non-ICMP flows only, use the same procedure as that of Section 1.5.1, but replace `PD_EQ=1` with `PD_EQ=0` in step 3. Alternatively, replace `$14==1` with `$14!=1` in step 2. Or if an entire flow file is preferred to the flow indexes only, set `PD_FORMAT=1` and replace `print $2` with `print $0` in step 2.