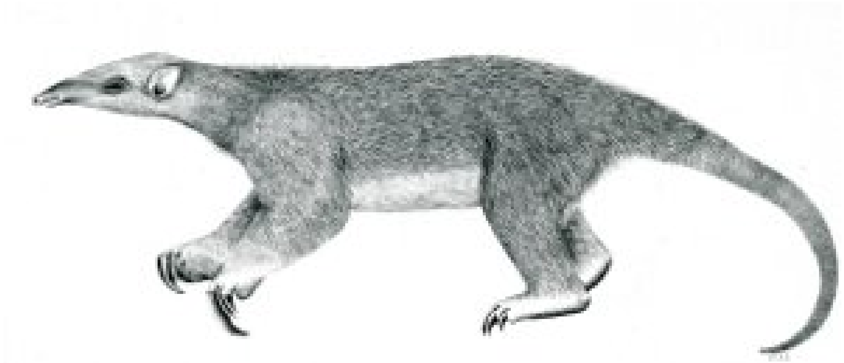
Tranalyzer2

Frequently Asked Questions

FAQ



Tranalyzer Development Team

Contents

1 FAQ

This section answers some frequently asked questions.

1.1 If the hashtable is full, how much memory do I need to add?

When T2 warns you that the hashtable is full, it also tells you how to correct the problem:

```
[INF] Hash Autopilot: main HashMap full: flushing 1 oldest flow(s)! Fix: Invoke T2 with
                              '-f 5' next time.
```

T2 calculates an estimate of the multiplication factor `HASHFACTOR` which you can set with the `-f` commandline option. By default the main hash autopilot is enabled which maintains the sanity of T2 even if it runs out of flow memory. Nevertheless, T2 will be faster if you feed him the recommended `-f` factor.

1.2 Can I change the timeout of a specific flow in my plugin?

That is possible because each flow owns a timeout value which can be altered even on packet basis. It enables the user to program stateful protocol plugins. Check out the `tcpStates` plugin as an inspiration.

1.3 Can I reduce the maximal flow length?

In *tranalyzer2/src/tranalyzer.h* you will find a constant called `FDURLIMIT`. Set it to the amount of seconds you like and T2 will terminate every flow with max `FDURLIMIT+1` seconds. And create a new flow for the next packet to come.

1.4 How can I change the separation character in the flow file?

The separation character is defined as `SEP_CHAR` in *utils/bin2txt.h*. It can be set to any character(s), e.g., `","` or `"||"`. In addition, the character(s) used for comments, e.g., column names, is controlled by `HDR_CHR` in the same file. Note that Tranalyzer default values are `"\t"` and `"%"`, respectively. Be advised that if you changed either of those values, some scripts may not work as expected.

1.5 How can I build all the plugins?

If you invoked the script setup.sh then you may use
`t2build -a`
otherwise, old school:

```
   cd /tranalyzer2-0.8.4
./autogen.sh -a
```

1.6 T2 failed to compile: What can I do?

If a dependency is missing, you should see an appropriate messsage, e.g., *Missing dependency libname*. If no such message is displayed, it could be that the Makefiles are outdated. Then use `autogen.sh -r` to force the rebuild of the Makefiles. A typical error requiring the use of `autogen.sh -r` is:

```
...
/bin/bash: line 10: automake-: command not found
Makefile:333: recipe for target `Makefile.in' failed
make[1]: *** [Makefile.in] Error 127
...
```

If you see the following message, then the autotools are not installed.

```
make: Entering directory '/home/user/tranalyzer2-0.8.4/tranalyzer2/doc'
make: Nothing to be done for 'clean'.
make: Leaving directory '/home/user/tranalyzer2-0.8.4/tranalyzer2/doc'
../autogen.sh: line 116: autoreconf: command not found
../autogen.sh: line 118: ./configure: No such file or directory

Failed to configure tranalyzer2
```

In this case, please refer to the *doc/tutorials/install.pdf*.

1.7 T2 segfaults: What can I do?

T2 never segfaults! Unless he deviates from his cosmic plan and indeed segfaults. The prominent reason are memory inconsistencies with old plugins being resident under `~/.tranalyzer/plugins/`.

1. Remove all the plugins: `rm ~/.tranalyzer/plugins/*.so`
2. Recompile the plugins, e.g., `cd ~/tranalyzer2-0.8.4/ && ./autogen.sh`
3. T2 should behave again.

For the developer:
If that does not fix the problem, recompile T2 in debug mode with `./autogen.sh -d` and try to run tranalyzer in *gdb*: `gdb -args ./tranalyzer -r file.pcap -w outpref`. If the error happens while writing flows, try to remove plugins until the error disappears. Finally, run the `segvtrack` script as follows: `segvtrack yourpcap`. This will automatically reduce the PCAP to the smallest set of packets which causes a segfault. If this does not help, send us a bug report at tranalyzer@rdit.ch with this pcap, T2 configuration (the values that differ from the default) and the plugins you are using. Then we will get a fix for you in no time.

1.8 socketSink plugin aborts with "could not connect to socket: Connection refused"

The `socketSink` plugins acts as a client in a socket communication. Therefore, a server listening to `SERVADD`, `DPORT` and `SOCKTYPE` is required. As described in the **Example** Section of the `socketSink` plugin documentation, a simple server can be set up with netcat as follows: `nc -l 127.0.0.1 6666`. Make sure the address and port match the values listed in *socketSink.h*.

1.9 T2 stalls after USR1 interrupt: What can I do?

It is a bug in the libpcap, which somehow is not thread-safe under certain conditions. Check whether T2 is set to default signal threading mode in (`main.h`):

- Set `MONINTTHRD` to 1
- Set `MONINTPSYNC` to 1

Do not forget to recompile T2 with `./autogen.sh` if you had to change the configuration.

Now the process of printing is detached from the packet capture and the output is synchronized to the packet processing main loop. Thus, pcap is never interrupted.

1.10 Can I reuse my configuration between different machines or Tranalyzer versions?

You can write a patch for `t2conf` and use it as follows: `t2conf --patch file.patch`. Revert the patch with the `--rpatch` option. The patch is a simple text file listing the defines to change, e.g., `IPV6_ACTIVATE <tab> 1 <tab> 0 <tab> tranalyzer2/src/networkHeaders.h`. For more details, refer to the documentation of `t2conf` .

1.11 How to contribute code, submit a bug or request a feature?

Contact the Anteater via email at tranalyzer@rdit.ch, and he will answer you.