Tranalyzer2

sqliteSink

SQLite

Tranalyzer Development Team

Contents

Copyright © 2008–2024 by Tranalyzer Development Team

1 sqliteSink

1.1 Description

The sqliteSink plugin outputs flows to a SQLite database.

1.2 Dependencies

1.2.1 External Libraries

This plugin depends on the **sqlite** library.

Ubuntu:	`sudo apt-get install`	`libsqlite3-dev`
Arch:	`sudo pacman -S`	`sqlite`
openSUSE:	`sudo zypper install`	`sqlite3-devel`
Red Hat/Fedora[1]**:**	`sudo dnf install`	`sqlite-devel`
macOS[2]**:**	`brew install`	`sqlite`

1.2.2 Core Configuration

This plugin requires the following core configuration:

- *$T2HOME/tranalyzer2/src/tranalyzer.h*:
  - `BLOCK_BUF=0`

1.3 Configuration Flags

The following flags can be used to control the output of the plugin:

Name	Default	Description
`SQLITE_OVERWRITE`	2	0: abort if table already exists 1: overwrite table if it already exists 2: append to table if it already exists
`SQLITE_HEX_AS_INT`	0	0: store hex numbers (bitfields) as text 1: store hex numbers (bitfields) as int
`SQLITE_TRANSACTION_NFLOWS`	40000	0: one transaction > 0: one transaction every *n* flows
`SQLITE_QRY_LEN`	32768	Initial length for query
`SQLITE_QRY_MAXLEN`	4194304	Maximal length for query
`SQLITE_DB_SUFFIX`	`".db"`	Suffix for the database name
`SQLITE_DBNAME`	`"/tmp/t2.db"`	Name of the database
`SQLITE_TABLE_NAME`	`"flow"`	Name of the table
`T2_SQLITE_SELECT`	0	Only insert specific fields into the DB
`SQLITE_SELECT_FILE`	`"sqlite-columns.txt"`	Filename of the field selector

[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

Copyright © 2008–2024 by Tranalyzer Development Team

Name	Default	Description
		(one column name per line)

1.3.1 Environment Variable Configuration Flags

The following configuration flags can also be configured with environment variables (`ENVCNTRL>0`):

- `SQLITE_QRY_LEN`
- `SQLITE_QRY_MAXLEN`
- `SQLITE_DB_SUFFIX`
- `SQLITE_TABLE_NAME`
- `SQLITE_SELECT_FILE`

1.3.2 Database Name

The database name is extracted from Tranalyzer input and/or `-w`/`-W` option. `SQLITE_DB_SUFFIX` is simply appended. Alternatively, an absolute path may be provided by uncommenting the `SQLITE_DBNAME` macro in *src/sqliteSink.h*.

1.4 Insertion of Selected Fields Only

When `T2_SQLITE_SELECT=1`, the columns to insert into the DB can be customized with the help of `SQLITE_SELECT_FILE`. The filename defaults to `sqlite-columns.txt` in the user plugin folder, e.g., *~/.tranalyzer/plugins*. The format of the file is simply one field name per line with lines starting with a '`#`' being ignored. For example, to only insert source and destination addresses and ports, create the following file:

```
# Lines starting with a '#' are ignored and can be used to add comments
srcIP
srcPort
dstIP
dstPort
```

1.5 Plugin Report Output

The following information is reported:

- Number of flows discarded due to main buffer problems

1.6 Example

```
# Run Tranalyzer
$ t2 -r file.pcap
# Connect to the SQLite database
$ sqlite3 file.db
# Number of flows
sqlite> select count(*) from flow;
# 10 first srcIP/dstIP pairs
```

Copyright © 2008–2024 by Tranalyzer Development Team

```
sqlite> select "srcIP", "dstIP" from flow limit 10;
# All flows from 1.2.3.4 to 1.2.3.5
sqlite> select * from flow where "srcIP" = '1.2.3.4' and "dstIP" = '1.2.3.5';
```

Copyright © 2008–2024 by Tranalyzer Development Team