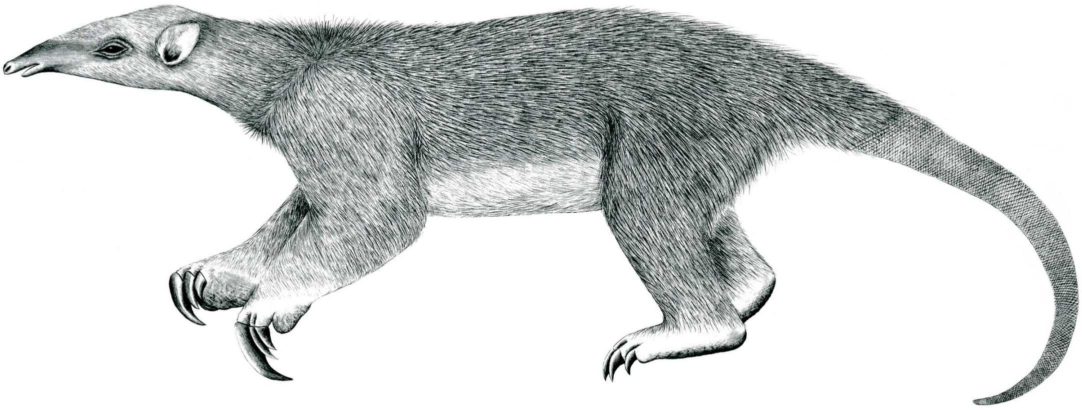
Tranalyzer2

mysqlSink

MariaDB / MySQL



Tranalyzer Development Team

Contents

1 mysqlSink

1.1 Description

The mysqlSink plugin outputs flows to a MariaDB / MySQL database.

1.2 Dependencies

1.2.1 External Libraries

This plugin depends on the **MariaDB** or **MySQL** library.

		MariaDB	**MySQL**
Ubuntu:	`sudo apt-get install`	`libmariadb-dev`	`libmysqlclient-dev`
Arch:	`sudo pacman -S`	`mariadb-libs`	
Gentoo:	`sudo emerge`	`mariadb-connector-c`	`mysql-connector-c`
openSUSE:	`sudo zypper install`	`libmariadb-devel`	
Red Hat/Fedora[1]**:**	`sudo dnf install`	`mariadb-connector-c-devel`	`community-mysql-devel`
macOS[2]**:**	`brew install`	`mariadb-connector-c`	

1.2.2 Core Configuration

This plugin requires the following core configuration:

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

1.3 Database Setup

1.3.1 Create a User with Create and Write Permissions

```
$ sudo mysql -u root mysql
...
MariaDB [mysql]> create user 'mysql'@'localhost' identified by 'mysql';
MariaDB [mysql]> grant all privileges on *.* to 'mysql'@''localhost' with grant option;
```

1.4 Configuration Flags

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

Name	**Default**	**Description**
`MYSQL_OVERWRITE_DB`	2	0: abort if DB already exists 1: overwrite DB if it already exists 2: reuse DB if it already exists

[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

Name	Default	Description
MYSQL_OVERWRITE_TABLE	2	0: abort if table already exists 1: overwrite table if it already exists 2: append to table if it already exists
MYSQL_TRANSACTION_NFLOWS	40000	0: one transaction > 0: one transaction every *n* flows
MYSQL_QRY_LEN	32768	Max length for query
MYSQL_HOST	"127.0.0.1"	Address of the database
MYSQL_DBPORT	3306	Port the DB is listening to
MYSQL_USER	"mysql"	Username to connect to DB
MYSQL_PASS	"mysql"	Password to connect to DB
MYSQL_DBNAME	"tranalyzer"	Name of the database
MYSQL_TABLE_NAME	"flow"	Name of the table
MYSQL_SELECT	0	Only insert specific fields into the DB
MYSQL_SELECT_FILE	"mysql-columns.txt"	Filename of the field selector (one column name per line)

1.4.1 Environment Variable Configuration Flags

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

- MYSQL_HOST
- MYSQL_DBPORT
- MYSQL_USER
- MYSQL_PASS
- MYSQL_DBNAME
- MYSQL_TABLE_NAME
- MYSQL_SELECT_FILE (require MYSQL_SELECT=1)

1.5 Insertion of Selected Fields Only

When MYSQL_SELECT=1, the columns to insert into the DB can be customized with the help of MYSQL_SELECT_FILE. The filename defaults to mysql-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
```

1.6 Example

```
# Run Tranalyzer
$ t2 -r file.pcap
# Connect to the MySQL database
$ mysql tranalyzer
# Number of flows
MariaDB [tranalyzer]> select count(*) from flow;
# 10 first srcIP/dstIP pairs
MariaDB [tranalyzer]> select "srcIP", "dstIP" from flow limit 10;
# All flows from 1.2.3.4 to 1.2.3.5
MariaDB [tranalyzer]> select * from flow where "srcIP" = ’1.2.3.4’ and "dstIP" = ’1.2.3.5’;
```