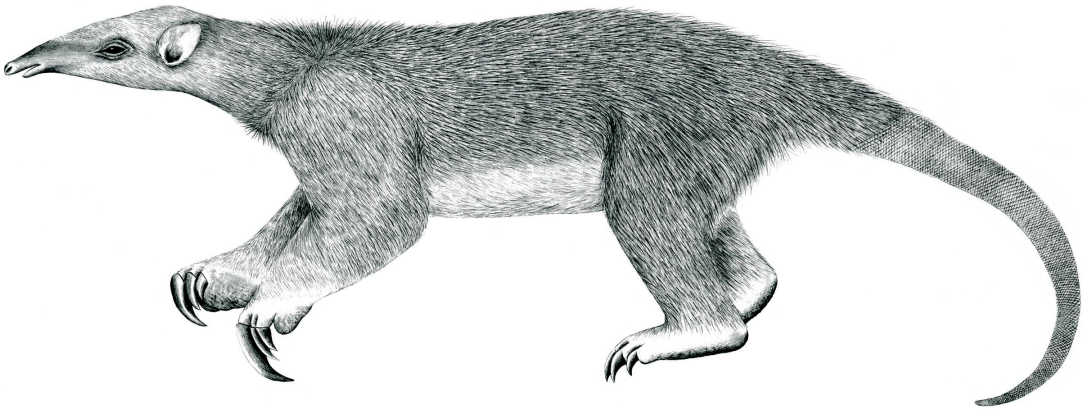
Tranalyzer2

mongoSink

MongoDB



Tranalyzer Development Team

Contents

1 mongoSink

1.1 Description

The mongoSink plugin outputs flows to a MongoDB database.

1.2 Dependencies

1.2.1 External Libraries

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

Ubuntu:	`sudo apt-get install`	`libmongoc-dev`
Arch:	`sudo pacman -S`	`mongo-c-driver`
Gentoo:	`sudo emerge`	`mongo-c-driver`
Red Hat/Fedora[1]**:**	`sudo dnf install`	`mongo-c-driver-devel`
macOS[2]**:**	`brew install`	`mongo-c-driver`

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
`MONGO_HOST`	`"127.0.0.1"`	Address of the database
`MONGO_PORT`	`27017`	Port the database is listening to
`MONGO_DBNAME`	`"tranalyzer"`	Name of the database
`MONGO_TABLE_NAME`	`"flow"`	Name of the database flow table
`MONGO_NUM_DOCS`	1	Number of documents (flows) to write in bulk
`MONGO_QRY_LEN`	2048	Max length for query
`MONGO_SELECT`	0	Only insert specific fields into the DB
`MONGO_SELECT_FILE`	`"mongo-columns.txt"`	Filename of the field selector (one column name per line)
`BSON_SUPPRESS_EMPTY_ARRAY`	1	Output empty fields
`BSON_DEBUG`	0	Print debug messages

[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`

1.3.1 Environment Variable Configuration Flags

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

- MONGO_HOST
- MONGO_PORT
- MONGO_DBNAME
- MONGO_TABLE_NAME
- MONGO_SELECT_FILE (require MONGO_SELECT=1)

1.4 Insertion of Selected Fields Only

When MONGO_SELECT=1, the columns to insert into the DB can be customized with the help of MONGO_SELECT_FILE. The filename defaults to mongo-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 Working with Timestamps (ISODate)

MongoDB stores timestamps in UTC as ISODate. To convert them to localtime, you may use the following query:

```
> db.flow.aggregate([{
      $project: {
          localTime: {
              $dateToString: {
                  date: "$timeFirst",
                  format: "%Y-%m-%d %H:%M:%S",
                  timezone: "Europe/Berlin"
              }
          }
      }
  }])
```

1.6 Example

```
# Run Tranalyzer
$ t2 -r file.pcap

# Connect to the Mongo database
$ mongosh tranalyzer
```

```
# Number of flows
> db.flow.countDocuments()

# 10 first srcIP/dstIP pairs
> db.flow.find({}, { _id:  0, srcIP: 1, dstIP: 1 }).limit(10)

# All flows from 1.2.3.4 to 1.2.3.5
> db.flow.find({ srcIP: "1.2.3.4", dstIP: "1.2.3.5" })
```

For examples of more complex queries, have a look in `$T2HOME/scripts/t2fm/mongo/`.

1.6.1 Clean up an existing database

```
# Connect to the Mongo database
$ mongosh tranalyzer

# Drop the database
> db.flow.drop()
```