
Nameles Documentation

Release latest

January 27, 2017

1 Overview	3
2 Getting Started	5
2.1 Install from source (with SQL port)	5
2.2 Install from source (without SQL port)	6
3 Usage	7
4 Method	9
5 NES Score	11
6 Total-Cost-of-Operation	13
7 Log File Format	15

First ever open source solution for detecting invalid traffic (ad fraud)

Overview

Nameles is an entropy based open source ad fraud detection solution that detects traffic anomalies agnostically across desktop and mobile. It has been proven[1] to deliver excellent results in detecting banner, video and app fraud at large scale with minimal Total-Cost-of-Ownership.

Learn more at <http://nameles.org>

Getting Started

At the moment you have recommended options:

2.1 Install from source (with SQL port)

<https://github.com/Nameles-Org/Nameles/tree/master/Nameles-SQL>

Create the environment:

```
cd entropy-postgresql && make && sudo make install && cd ..
```

Create a database:

```
psql -f create_db.sql
```

Create the entropy functions in the new database:

```
psql -d nameless -f entropy-postgresql/create_functions.sql
```

Install the required python packages for `log_migration.py` to upload compressed log files:

```
pip install psycopg2
pip install tld
```

Upload your logfile format:

```
./nameless-log-migration <logday> /path/to/log/files/my_log_000*.csv.gz
```

Where `<logday>` is the date of the file you are uploading.

NOTE: you can test your installation first using the provided test log file:

```
wget https://github.com/Nameles-Org/Nameles/blob/master/tests/ssp_bid_sample.csv.gz && ./nameless-log-
```

In some cases you have to do minor changes to your log file format before running `nameless-log-migration` script.

This will create the following tables in the database:

`tuples.ip_ref_<logday>` Table of tuples `<IP, referrer, count>` with the aggregate count of non-concurrent ad-requests with the same IP and referrer.

Temporary tables with the total number of ad-requests and normalized entropy score for the log day of both, referrers and IPs. These tables will be merged in the stats tables for IPs and referrers.

2.2 Install from source (without SQL port)

<https://github.com/Nameles-Org/Nameles/tree/master/Nameles-nodb>

NOTE: this will still require PostgreSQL (9.4) to be installed in the system.

Install from Linux Package [EXPERIMENTAL]

<https://github.com/Nameles-Org/Nameles/tree/master/Easy%20Install%20Option>

Usage

By Querying the Database

The tables in the database can be queried as regular SQL tables.

Through an API

Method

Nameles use a three staged approach where:

1. Entropy is computed for each referrer in the dataset
2. The Normalized Entropy Score (NES) is calculated for each referrer
3. Based on NES the referrers are grouped in to statistically meaningful groups

The grouping in step-3 is done based on how suspicious the referrer is based on its NES score. The groupings:

critical (highest level of suspicion)

high

moderate

low (highest level of suspicion)

The way the score is calculated and the resulting grouping is affected by the way the system owner sets the rules for scoring. The owner is left with the decision if they want to set the system to have many false positive or many false negatives. The recommendation is for “paranoid” setting, as there is much more inventory in the programmatic market than there is demand. In fact, due to the high perishability of media inventory, it is likely that more than 90% of all inventory never gets sold. Even if just 10% is never getting sold, filtering out 10% would not reduce inventory that is available for buying.

NES Score

The unique value of Nameles is the Normalized Entropy Score (NES) it provides for any referrer, regardless if it's desktop or mobile, app, video or banners. The formal expression to compute NES is as follows:

Total-Cost-of-Operation

One of the key considerations in the research and development that led to Nameles was Total-Cost-of-Onwership. This can be split in to two parts; Non-Recurring-Cost (NRC) and Recurring-Cost (RC).

In the case of NRC, we can safely say that it is negligible. In fact, a small startup company could fully deploy and adopt for actual business use, the Nameles system with 1 man-week of work, if not less. In the case of bigger organizations, there tends to be more overhead with introducing anything new, so NRC might be higher depending on the culture and other factors pertaining to the organization in question.

In terms of RC, depending on the database backend, the C codes that compute the NES can process roughly 10 billion rows of bid stream data per 24 hour period with a 16-core 48gb RAM Linux server. Using the same machine and the SQL provided you'll find in this repository, scores can be computed to up to 2 billion rows per day.

Log File Format

Nameles has been built and tested with various common bid / log file formats, and can be easily setup to work with others. The codes that you find in the repo will work with the following format out-of-the-box:

```
107.219.252.241,LF5rUhJzH15Oh9BrtFRRiu7cY+g=,"Mozilla/5.0 (Linux; Android 6.0.1; SM-G900V
Build/MMB29M; wv) AppleWebKit/537.36 (KHTML, like Gecko) Version/4.0 Chrome/54.0.2840.85
Mobile Safari/537.36",net.zedge.android,,2016-12-01 14:19:04
```

NOTE: this row highlights example of one row / entry in a larger file.