



Firefox Origin Telemetry with Prio

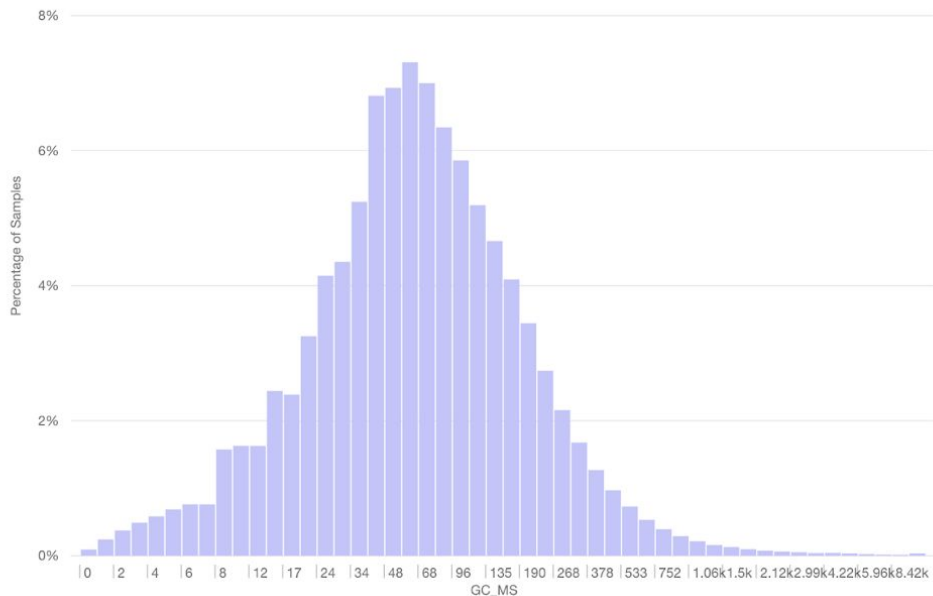
2020-08-15

Anthony Miyaguchi
Data Engineer

Data at Mozilla

GC_MS distribution for Firefox Desktop nightly 71, on any OS (51) any architecture (3) with any process and compare by none

Time spent running JS GC (ms) ⓘ More details



Histogram Type exponential
Ping Count 4.36M
Sample Count 794.22M
Sample Sum 163.25B
Number of dates 17
Selected Dates 2019/09/02
to
2019/09/18

5th Percentile 8.94
25th Percentile 33.93
Median 65.09
75th Percentile 130.72
95th Percentile 386.39

Notice percentiles are estimated based on values in the histogram. Values are only guaranteed to be accurate to the nearest bucket.

Export CSV

Export JSON

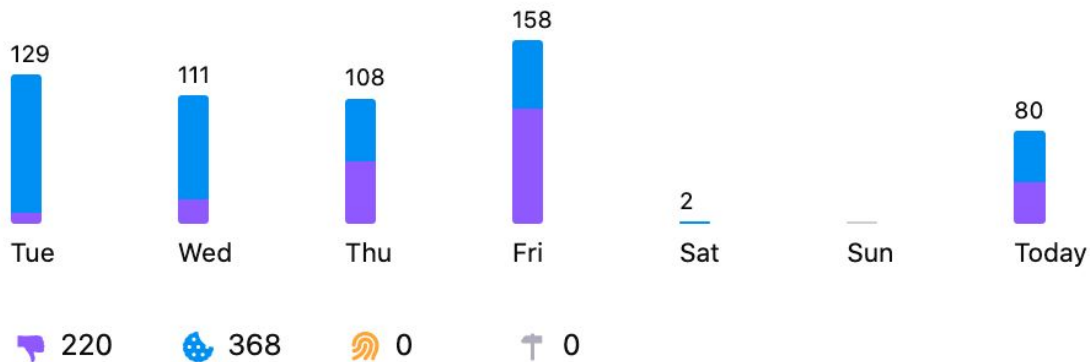
Content Blocking



Enhanced Tracking Protection: Always On

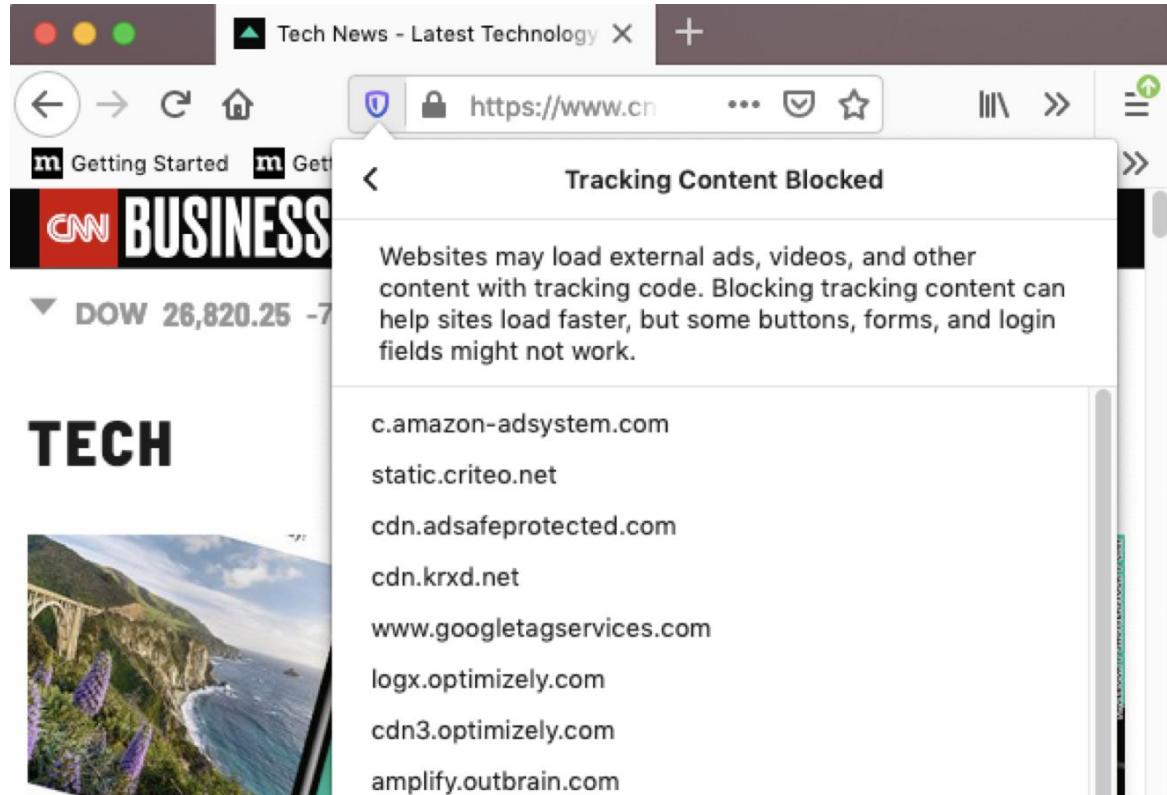
Nightly automatically stops companies from secretly following you around the web.

Nightly blocked 588 trackers over the past week



about:protections

How many times has `example.com` been blocked?

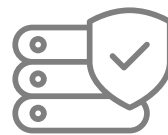


Prio

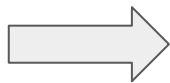
A system for privacy preserving aggregates

Private	Clients are anonymous if at least one server is honest
Robust	Malicious clients cannot corrupt the resulting output
Scalable	Secret-shared non-interactive proofs

Scheme for Private Sums



$(13, -8, -4)$



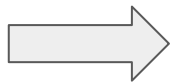
(13)

(-8)

(-4)



$(-4, -2, 7)$



$(13-4)$

$(-8-2)$

$(-4+7)$



(\dots)

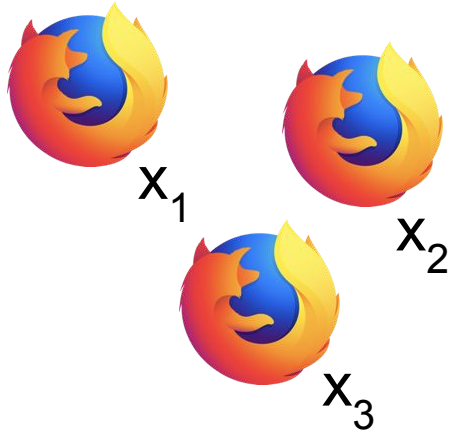


$(13-4+\dots)$

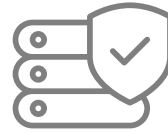
$(-8-2+\dots)$

$(-4+7+\dots)$

Scheme for Private Sums



$$(13-4+\dots)$$



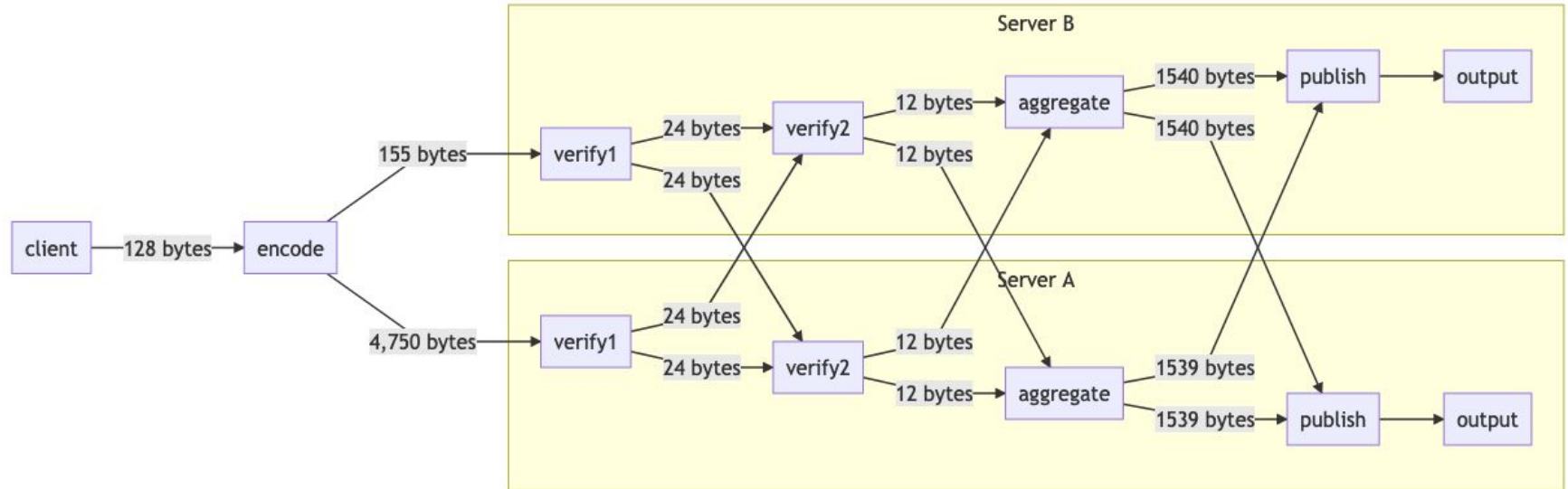
$$(-8-2+\dots)$$



$$(-4+7+\dots)$$

$$= x_1 + x_2 + x_3$$

Data Flow



Pilot Program and Verification

libprio - A Prio library in C using NSS

Warning: We do our best to write bug-free code, but I have no doubt that there are scary bugs, side-channel attacks, and memory leaks lurking herein.

Security bugs: If you find a security-critical bug in libprio, please report it to Mozilla using the contact information on [this page](#).

Verifying Correctness

reason	prio_control	prio_observed	prio_diff
shutdown	[14891, 1578, 1]	[14891, 1578, 1]	[0, 0, 0]
shutdown	[10829, 3160, 1]	[10829, 3160, 1]	[0, 0, 0]
shutdown	[18333, 2939, 3]	[18333, 2939, 3]	[0, 0, 0]
aborted-session	[467, 155, 0]	[459, 150, 0]	[8, 5, 0]
aborted-session	[470, 129, 0]	[465, 129, 0]	[5, 0, 0]
aborted-session	[265, 49, 0]	[261, 49, 0]	[4, 0, 0]
daily	[1067, 471, 0]	[0, 0, 0]	[1067, 471, 0]
daily	[935, 325, 0]	[0, 0, 0]	[935, 325, 0]
daily	[1566, 934, 1]	[0, 0, 0]	[1566, 934, 1]
environment-change	[1923, 500, 0]	[0, 0, 0]	[1923, 500, 0]
environment-change	[1411, 697, 0]	[0, 0, 0]	[1411, 697, 0]
environment-change	[2100, 684, 0]	[0, 0, 0]	[2100, 684, 0]

Firefox Origin Telemetry and Validation

current data ▾

Home

General Data

Environment Data

Session Information

Scalars

Keyed Scalars

Histograms

Keyed Histograms

Events

Raw JSON

Origin Telemetry

Find in Origin Telemetry

[Firefox Origin Telemetry](#) encodes data before it is sent so that Mozilla can count things, but not know whether or not any given Firefox contributed to that count. ([learn more](#))

content.blocking_blocked_TESTONLY

origin	count
outbrain.com	3
amazon-adsystem.com	3
rubiconproject.com	3
agkn.com	2
facebook.com	3
criteo.com	3
imrworldwide.com	3
adnxs.com	3
adservice.google.com	3
krxd.net	3
bounceexchange.com	3

“prio” ping

This ping transmits [Origin Telemetry](#) data.

The client id is not submitted with this ping. The [Telemetry Environment](#) is not submitted in this ping.

```
{
  "type": "prio",
  ... common ping data
  "payload": {
    "reason": {periodic, max, shutdown}, // Why the ping was submitted
    "prioData": [{
      encoding: <encoding name>, // Name of App Encoding applied. e.g. "content-blocking-1"
      prio: {
        // opaque prio-specific payload. Like { a: <base64 string>, b: <base64 string> }
      },
    }, ... ],
  }
}
```



mozilla/prio-processor ☆

↓ Pulls 1.0K

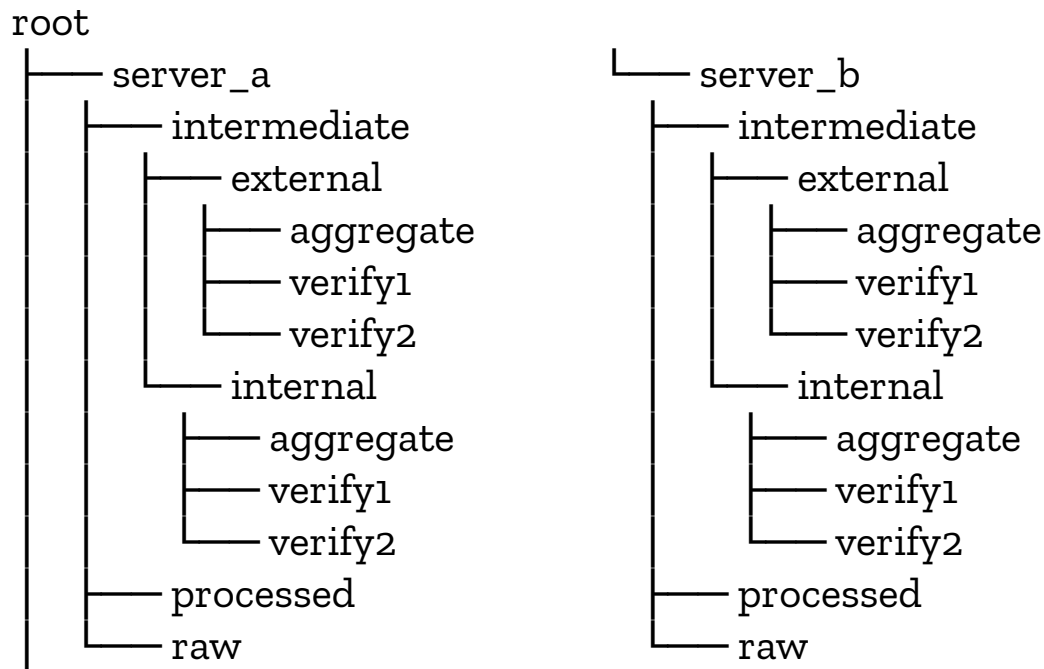
By mozilla • Updated 7 days ago

A python interface into libprio

Container

- Spark ETL to prepare Prio pings from Mozilla ingestion (BigQuery, GCS)
- Scripts for idempotent batch processing using S3-compatible object store
- Deployed as a docker image and configured via environment variables e.g.
`docker run mozilla/prio-processor:latest`

Interoperability via Cloud Object Storage



On **DAG: prio_processor**

schedule: 1 day, 0:00:00

Graph View

Tree View

Task Duration

Task Tries

Landing Times

Gantt

Details

Code

Refresh

Delete

success

Base date:



2019-09-30 00:02:56+00

Number of runs:

25

Run:

scheduled__2019-09-30T00:02:55.575489+00:00

Layout:

Left->Right

Go

Search for...

GoogleCloudStorageToGoogleCloudStorageOperator

PythonOperator

SubDagOperator

success

running

failed

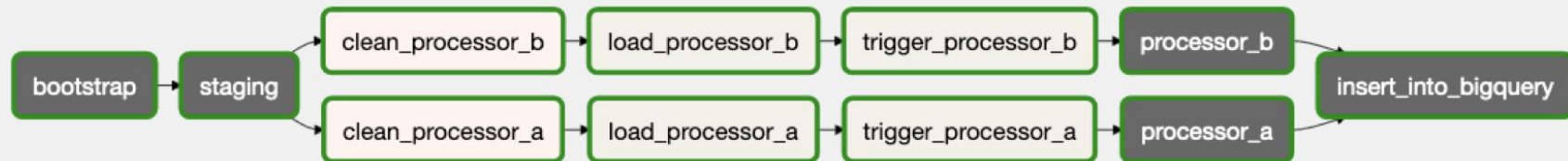
skipped

rescheduled

retry

queued

no status



submission_date	origin	AGGREGATE	origin_rank	pageloads	normalized
2020-07-12	youtube.com	3	1	26	0.12
2020-07-12	__UNKNOWN__	2	2	26	0.08
2020-07-12	doubleclick.net	2	3	26	0.08
2020-07-12	quantserve.com	2	4	26	0.08
2020-07-12	accounts.google.com	1	5	26	0.04

Current State

- Collection is enabled on Firefox Nightly (opt-in) for ~2500 block list rules
- In the process of engaging with other organizations for running another server
- libprio Python bindings deployed to PyPi (pip install prio)
- Spark backend implementation in prio-processor

Links

[Paper] [Prio: Private, Robust, and Scalable Computation of Aggregate Statistics](#)

[Blog] [Testing Privacy-Preserving Telemetry with Prio](#)

[Blog] [Firefox Origin Telemetry: Putting Prio in Practice](#)

[Blog] [Next steps in privacy-preserving Telemetry with Prio](#)

[GitHub] [mozilla/libprio](#)

[GitHub] [mozilla/prio-processor](#)

[Firefox Source Docs] [Origin Telemetry](#)