WhoTracks.me Dataset - Variable Descriptions

Karlo Lukic, karlo.lukic@pm.me

01.05.20

This document represents a summary of WhoTracks.me data compiled from paper readings, blog posts and official GitHub's variable descriptions.

Data collection

Data was collected from May 2017 from users that used Cliqz browser extension. In Feb 2018, 70% of the data came from German users according to this blog post. Then in March 2018, users of Ghostry FireFox extension – and Ghostry extension available for other browsers (Safari, Chrome, Opera and Edge) from users that opted-in to *HumanWeb* data collection – were added to the dataset. This caused a slight decrease in the avg. no. of trackers in April 2018, since Ghostry users were blocking more trackers. This is explained in this and this blog posts.

This blog post illustrates where the traffic came from in April 2018: Germany and USA being most representative.

This blog post notes that WhoTracks.me does not collect data for pages with no trackers; in other words, collected data for all sites contains some number of third-parties and tracking.

Datasets

There are 5 main datasets on WhoTracks.me GitHub's repo – unlike 4 datasets available in "Explorer" section on the website:

- `sites.csv`: Stats for number of trackers seen on popular websites.
- `site_trackers.csv`: Stats for each tracker on each site.
- `domains.csv`: Top third-party domains seen tracking.
- `trackers.csv`: Top trackers - this combines domains known be operated by the same tracker.
- `companies.csv`: Top companies - aggregates the stats for trackers owned by the same company.

> "We structure each subsection in a way that describes measurements in the perspective of the parties involved: websites, third parties and users. This enables us to better measure the dynamics of the industry."

In addition, WhoTracks.me team provides up-to-date SQL database that links third-party domains to trackers, which are then linked to unique companies operating those trackers. This is similar to Disconnect's Tracker List and webXray's Domain Owner List, yet much more comprehensive. These parties are already categorized accordingly within the datasets.

Variable descriptions

All 5 above datasets share similar aggregated variables. The difference therefore, lies in the *perspective* of each dataset. Here are the variable descriptions ("contexts" are added to variables for groupings):

General context:

- site - one of the most frequently visited websites from a proportion of traffic in certain month. That means that the most popular websites in the dataset were not generated by Alexa or similar services, but by real users. Total number of published most-visited-user-generated sites increased over time: e.g. from 700 most visited sites in 2017 to over 1000 sites in 2020 in Global datasets. Note: average monthly traffic (page loads) of users was around 100 million page loads during 2017, and increased to 300-500 million page loads from April 2018 onward (as described in this part of WhoTracks.me paper). String.
- month - month of observation. Global traffic data starts from May 2017 and ends with latest GitHub release; EU/US traffic split starts from April 2018 and ends with latest GitHub release. mm-yyyy format string/date.
- country - main region where the traffic is coming from: e.g. global, US, EU, DE, FR. String.
- category - site's category (in sites.csv). String.
- tracker_category - tracker's category (in sites_trackers.csv). Descriptions of tracker categories are provided here. String.
- popularity - the relative amount of traffic compared to the most popular site (described here). Float between 0 and 1.

Utilised tracking context (stateful) – generates more persistant tracking ID by trackers:

- cookies - proportion of pages where a cookie was sent by the browser, or a Set-Cookie header was returned by the tracker's server. Float between 0 and 1.

Utilised tracking context (stateless) – generates less persistant tracking ID by trackers:

- bad_qs - proportion of pages where a unique identifier (UID) was detected in the query string parameters sent with a request to this tracker. More on this here. The methodology for this detection can be found in the paper. Float between 0 and 1.

  "Note that these detection methods assume that trackers are not obfuscating the identifiers they generate."

Utilised tracking context (stateful + stateless) – either cookie tracking or fingerprinting context, inclusive:

- tracked - proportion of pages where a UID transmission was detected, either via cookies or bad_qs. Float between 0 and 1.

  "We define tracking as when a service is able to collect and correlate data across multiple sites."

Secure context – tracker used HTTPS requests – instead of HTTP – to load its content:

- `https` - proportion of pages where the tracker only used `HTTPS` traffic. Float between 0 and 1.

Tracking requests context – we report the mean number of third-party requests per page for each tracker, and the subset of these requests in a tracking context:

- `requests` - average number of requests made to the tracker per page. Positive float.
- `requests_tracking` - average number of requests made to the tracker with tracking (cookie or query string) per page. Positive float.

Tracking cost context – how much page loads do trackers clog up by being loaded, more on this in the Tracker Tax paper:

- `content_length` - average of `Content-Length` headers received per page. This is an approximate measure of the bandwidth usage of the tracker. Expressed in bytes. Positive float.

  > "As users navigate the web, they load content from websites they visit as well as the third parties present on the website. ... Previous studies have found that each extra third party added to the site will contribute to an increase of 2.5% in the site's loading time."

Tracker's blocking context – how often the tracker is affected by blocklist-based blockers:

- `requests_failed` - average number of requests make to the tracker per page which do not succeed. In other words, avg. number of failed requests per page load (for comparison with `requests` to get an idea of how aggressive the blocking is). This is an approximate measure of blocking from external sources (i.e. adblocking extensions or firewalls). Measure added in Dec 2017. Positive float.
- `has_blocking` - proportion of pages where some kind of external blocking of the tracker was detected.Measure added in Dec 2017. Float between 0 and 1.

  > "These signals [`requests_failed` and `has_blocking`] should be able to tell us something about the impact of blocking on different trackers in the ecosystem. For example, we see evidence of blocking 40% of the time for Google Analytics and Facebook [in Dec 2017], and between 10% and 20% of requests failing. Thus, anyone using these services to measure activity and conversions on their sites must reckon with error rates in these orders. We also can see how new entrants can initially avoid the effects of blocking - for Tru Optik and Digitrust who we mentioned earlier, we measure only 5 and 1% of pages which may be affected by blocking."

Tracker's content loading context – proportion of page loads where specific resource types were loaded by the tracker (e.g. scripts, iframes, plugins)

Signals for the frequency with which certain resource types are loaded by third-parties (measures added in Feb 2018):

- `script` - JavaScript code (via a `<script>` tag or web worker).

  > "If a third-party Javascript file is loaded into the page, the third-party is given the ability to modify the page at will, intercept all user input on the page, as well as load any other scripts or third parties they wish. ... any third-party which is permitted to load Javascript in the login document will have to ability to read users' login information inputted into this page."

- `iframe` - a subdocument (via `<frame>` or `<iframe>` elements).

  > "Content loaded into an iFrame context is safer, as this is a sandboxed environment."

- `beacon` - requests sent through the Beacon API. More on this here:

  > "A tracking pixel, is one of various techniques used on web pages or email, to unobtrusively (usually invisibly) allow checking that a user has accessed some content."

- `image` - image and imageset resources.

  > "Loading third-party images in the page allows the third-party to know the page you're visiting, via the Referer header, your IP address, and may allow them to further track your browser via Cookies"

- `stylesheet` - CSS files.
- `font` - custom fonts.
- `xhr` - requests made from scripts via the XMLHttpRequest or fetch APIs.
- `plugin` - requests of `object` or `object_subrequest` types, which are typically associated with browser plugins such as Flash.
- `media` - requests loaded via `<video>` or `<audio>` HTML elements.

  > "By reporting these [above resource types] values we can further characterize tracker behaviours, and quantify risks, such as which trackers are being permitted to load scripts on certain pages. With this data we can see that, for example, Google Analytics loads their script on each page load (98% of the time), then registers the visit via a pixel on 59% of page loads. We also see that on 6% of pages a request is also made via the Beacon API. Similarly, if we look at the Webtrekk tracker, which is present on many popular German websites, we can see that on sensitive websites such as banking (dkb.de) and health insurance (tk.de), the tracker is loaded without scripts. This is at least an indication that in certain contexts website owners are taking care to minimise the potential risk of a third-party being compromised and gathering sensitive information from the page, or even collecting sensitive information by mistake."

Tracker's presence context – there are also counts of presences of other entities in the aggregation. This enables us to, for example, count how many of a tracker's domains they use simultaneously on average, or how many different trackers and companies are usually present on sites:

- `hosts` - avg. number of tracker's domains present on site. Several domains are grouped under `trackers`, e.g.: `facebook.com` and `facebook.net`grouped under `facebook` tracker. Positive float.
- `trackers` - avg. number of trackers present on site. Trackers are grouped under `companies`, e.g.: `facebook`, `facebook_cdn`, `facebook_graph`, `...`, grouped under Facebook. Positive float.

  > "We define a 'tracker' as a third-party domain which is: a) present on multiple (> 10) different websites with a significant combined traffic, b) uses cookies or fingerprinting methods in order to transmit user identifiers"

- `companies` - avg. number of companies present on site. Positive float.

Tracker reach context – for domain, trackers and companies aggregations, there are two extra measures:

- `reach`: Proportional presence across all page loads (i.e. if a tracker is present on 50 out of 1000 page loads, the reach would be 0.05). Value is a float between 0 and 1.

  > "We define a tracker or company's 'reach' as the proportion of the web in which they are included as a third-party."

- `site_reach`: Presence across unique first party sites. e.g. if a tracker is present on 10 sites, and we have 100 different sites in the database, the site reach is 0.1. Value is a float between 0 and 1.

  > "Alternatively, we can measure 'site reach', which is the proportion of websites (unique first-party hostnames) on which this tracker has been seen at least once."

Note: This measure was redefined in Feb 2019 as `site_reach_top10k`: the number of sites in the top 10,000 which have this tracker on more than 1% of page loads" according to this blog post. A further value, `site_avg_frequency` gives the mean presence across these sites. Positive floats.

> "Given that the top 10,000 sites account for 75% of page loads in our data, we decided to measure the presence across this fixed set of sites."

Differences between `reach` and `site_reach` (according to above blog post):

- High reach and high site reach - Ubiquitous presence across both popular and less popular sites; A common example of that would be Google Analytics.
- High reach and low site reach - Present primarily on few popular, high-traffic sites; One such example would be Wikimedia, which, due to Wikipedia's popularity, is loaded very often (hence high reach), but present on few sites resulting in a low(er) site reach. Another example, for similar reasons, would be Ebay Stats,
- Low reach and high site reach - Only appearing rarely on many sites, e.g. only on a small number of pages for each site; In this category appear extensions that operate as "man in the middle", such as Kaspersky Labs.
- Low reach and low site reach - Present on few lower-traffic sites. This includes smaller trackers.

Additional measures without explanation (todo: contact Sam)

- `referer_leaked` - unexplained (my best guess: proportion of total page loads in which the HTTP referer header was transmitted to a tracker). Float between 0 and 1. todo: double-check this

  > "The Referer request header contains the address of the previous web page from which a link to the currently requested page was followed."

- `referer_leaked_header` - unexplained (my best guess: proportion of total page loads in which the HTTP referer header was transmitted to a tracker while including full URL of the previously visited web page). Float between 0 and 1. todo: double-check this

- `referer_leaked_url` - unexplained (no best guess). todo: double-check this

- `cookie_samesite_none` - unexplained (no best guess). todo: double-check this

- `t_active` - unexplained (no best guess). todo: double-check this

Added variables

These are the variables I added for the DID analysis project:

- `category_fg` - one of 55 website categories provided by Fortiguard's Web Filter.
- `ipv4` - first IPv4 address of a site (server's IP address), derived from NSLookup via R package curl
- `server_continent_name` - continent of IPv4 (server's geolocation) according to latest edition of DB-IP "IP to City Lite" MMDB database. Categorical string.
- `server_country_name` - country of IPv4 (server's geolocation) according to latest edition of DB-IP "IP to City Lite" MMDB database. Categorical string.
- `server_country_code` - country code of IPv4 (server's geolocation) according to latest edition of DB-IP "IP to City Lite" MMDB database. Categorical string.
- `is_treated` - indicator variable specifying if the site is subjected to GDPR regulation (`1`) or not subjected to GDRP regulation (`0`) according to defined control/treatment framework. For main sample, includes 28 EU member states, 3 EEA regions and 9 outermost regions, similar to this encoding but excluding Montenegro (according to Article 3.3.). Boolean integer.
- `time` - indicator variable specifying pre- (`0`) and post-GDPR (`1`) periods. Boolean integer.
- `duplicate_site` - indicator variable specifying if the website in the sample has 1 (`0`) or 2 (`1`) observations: i.e. website with 2 observations was visited by both EU and US users, while website with 1 observation was visited either by US or EU users. Boolean integer.