WhoTracks.Me: Shedding light on the opaque world of online tracking

Arjaldo Karaj
arjaldo@cliqz.com

Sam Macbeth
sam@cliqz.com

Rémi Berson
remi@cliqz.com

Josep M. Pujol
josep@cliqz.com
Cliqz GmbH
Arabellastraße 23
Munich, Germany

ABSTRACT

Online tracking has become of increasing concern in recent years, however our understanding of its extent to date has been limited to snapshots from web crawls. Previous attempts to measure the tracking ecosystem, have been done using instrumented measurement platforms, which are not able to accurately capture how people interact with the web. In this work we present a method for the measurement of tracking in the web through a browser extension, as well as a method for the aggregation and collection of this information which protects the privacy of participants. We deployed this extension to more than 5 million users, enabling measurement across multiple countries, ISPs and browser configurations, to give an accurate picture of real-world tracking. The result is the largest and longest measurement of online tracking to date based on real users, covering 1.5 billion page loads gathered over 12 months. The data, detailing tracking behaviour over a year, is made publicly available to help drive transparency around online tracking practices.

Categories and Subject Descriptors

K.4 [**COMPUTERS AND SOCIETY**]: Privacy

Keywords

Online Tracking, Privacy by design, Open Data

1. INTRODUCTION

On the modern web our actions are monitored on almost every page we visit by third-party scripts which collect and aggregate data about users' activities and actions. A complex and dynamic ecosystem of advertising and analytics has emerged to optimise the monetization of this data, and has grown to such an extent that 77% of pages the average user will visit contain trackers [19], and with individual trackers present on over 60% of the top 1 million sites [11].

Monitoring this ecosystem has been the focus of recent efforts, looking into the methods used to fingerprint users and their devices [25], and the extent to which these methods are being used across the web [5], and quantifying the value exchanges taking place in online advertising [7, 27]. There is a lack of transparency around which third-party services are present on pages, and what happens to the data they collect is a common concern. By monitoring this ecosystem we can drive awareness of the practices of these services, helping to inform users whether they are being tracked, and for what purpose. More transparency and consumer awareness of these practices can help drive both consumer and regulatory pressure to change, and help researchers to better quantify the privacy and security implications caused by these services. With the EU's General Data Protection Regulation imminent at the time of writing, monitoring will be important to help detect violations.

Most previous work on measuring tracking prevalence at scale has focused on the engineering of crawlers which emulate a web browser visiting a series of pages [11, 21]. These systems instrument the browser to collect detailed information about each page loaded. This method can scale well, however, bias is introduced by the choice of crawling platform, the physical location from which the crawl is run, and the sites chosen to be crawled. Further limitations exist around getting data from pages behind authentication walls, such as in online banking portals, e-commerce checkout pages, paywalled content, and 'walled gardens' like Facebook and LinkedIn. Lastly, these crawls capture an instantaneous state of the ecosystem, but do not enable longitudinal analysis. Longitudinal studies have typically been done on a smaller scale to one-off crawls [18, 17].

This work contributes a system for the continuous measurement of the presence of third-parties across the web, and the tracking methods employed. This system gathers measurements via a large population of users who consent to data collection via a browser extension. We deploy a monitoring mechanism which collects data on third-party trackers for pages users visit, and employ a privacy-by-design methodology to ensure potential identifiable data or identifiers are removed on the client side before transmission. This enables measurement of tracking as observed by real users during normal browsing activities, at scale, across multiple browsers and physical locations, while respecting the privacy of the users collecting the data. This overcomes may of the issues encountered by crawl-based analyses of tracking.

Our method,using instrumented browsers distributed to users who consent to gathering data during their normal browsing activity can achieve a greater scale than crawling. In previous work, we analysed 21 million pages loaded by 200,000 users in Germany [30], and analysis of data collected from Ghostery's GhostRank covered 440 million pages for 850,000 users [19]. In this paper we present the *WhoTracks.Me* dataset, which contains aggregated data on third-party presence and tracking, released monthly. The data is generated by Ghostery and Cliqz users who have consented to anonymized HumanWeb [22] data collection. This generates data on an average of 100 million page loads per month, increasing to over 300 million since April 2018, and currently spans 12 months[1].

This paper is organised as follows. In Section 2 we describe how online tracking can be measured at scale during normal browser usage. We also describe common tracking methods and how they can be detected using browser extension APIs. In Section 3 we outline our approach to collection of the page load data, and how we prevent this data from being deanonymizable. Section 4 covers how we aggregate the collected data and generate meaningful statistics to describe the tracker ecosystem. We also describe our database which maps over 1000 tracker domains to services and companies which operate them. A selection of results are presented in Section 5, which show the extent of tracking which we have measured from 12 months of data, from a total of 1.5 billion page loads.

The work makes the following contributions:

- The largest longitudinal study of online tracking to date, in terms of number of pages and sites analysed, with a total of 1.5 billion pages analysed, and data on around 950 trackers and 1300[2] popular websites published under a permissive Creative Commons license.
- A public data set containing aggregated statistics on trackers and websites across the web.
- An open database to attribute common third-party domains to services and companies, containing over 1000 tracker entries.
- A method and implementation of a system for measuring tracking context in the browser, including fingerprinting detection based on [30].
- A system for the collection of the measured page load data which safeguards the privacy of the users from whom the data originates by removing or obfuscating any potential identifiable information in individual messages, and removing data which could be used to link messages together.
- A website providing information based on the collected data for interested users, and containing educational resources about online tracking.
- Results, reproducing findings of previous tracking studies, showing trends in online tracking over the last year, and providing new insights on previously unmeasured tracking.

2. MEASURING ONLINE TRACKING

Online tracking can be characterised as the collection of data about user interactions during the course of their web browsing. This can range from simply recording which types of browser access a particular page, to tracking all mouse movements and keystrokes. Of most concern to privacy researchers is the correlation and linkage of the data points from individual users across multiple web pages and web sites, primarily because of the privacy side-effects this entails: such histories, linked with identifiers, even when pseudo-anonymous, can be easily associated with individuals to whom they belong [28].

In this work we aim to measure the extent of this latter kind of tracking: the collection of linkable data points which generate a subset of users' browsing histories. As with other studies [11, 21, 17, 6], we do this by instrumenting the browser to observe the requests made from each page visited, and looking for evidence of identifiers which could be used to link messages together. Unlike other studies, which generally set up automated crawls to popular domains, we deploy our probe to users of the Cliqz and Ghostery browser extensions. This gives several advantages:

- *Scale*: The probe is deployed to over 5 million users, which gives us up to 350 million page load measurements per month. Such scale cannot practically be achieved with crawlers.
- *Client diversity*: With over 5 million users, we can obtain measurements from a myriad of network and system environments. This includes network location, ISP, Operating System, browser software and version, browser extensions and third-party software. All of these factors may have some influence on observed tracking. Previous studies using crawling suffer from a monoculture imposed by tooling limitations: Firefox on Linux in an Amazon data-centre.
- *The non-public web*: Stateless web crawling limits one's access to the public web only. These are pages which are accessible without any login or

[1] May 2017 to April 2018

[2] We intend to increase these numbers as our database grows.

user-interaction required. This excludes a significant proportion of the web were tracking occurs, such as during payments on E-commerce sites, when accessing online banking, or on ‘walled-gardens’ such as Facebook [16].

The downside of this approach is that when collecting data from real users as they browse the web, there could be privacy side-effects in the data collected. The aim is to be able to measure the extent of tracking, but without collecting anything which could identify individuals, or even having any data value that someone may consider private. Therefore, great care must be taken in the collection methodology: what data can and cannot be collected, and how to transmit this privately. Due to these constraints, the data we can collect is of much lower resolution as what can be collected from crawling. Therefore these two approaches can complement each other in this regard. We describe our methodology of privacy-preserving data collection in this paper.

2.1 Tracking: a primer

Tracking can be defined as collecting data points over multiple different web pages and sites, which can be linked to individual users via a unique user identifier. The generation of these identifiers can be *stateful*, where the client browser saves an identifier locally which can be retrieved at a later time, or *stateless*, where information about the browser and/or network is used to create a unique fingerprint. In this section we summarise the common usage of these methods:

2.1.1 Stateful tracking

Stateful tracking utilises mechanisms in protocol and browser APIs in order to have the browser save an identifier of the tracking server’s choosing, which can be retrieved and sent when a subsequent request is made to the same tracker.

The most common method is to utilise browser cookies. As this mechanism is implemented by the browser, it is a client-side decision whether to honour this protocol, and how long to keep the cookies. Almost all browsers offer the option to block cookies for third-party domains when loading a web page, which would prevent this kind of tracking. However, browsers have defaulted to allow all cookies since the cookie specification was proposed, leading to many services and widgets (such as third-party payment and booking providers) relying on third-party cookies to function.

Other stateful methods include the JavaScript `localStorage` API [4], which enables Javascript code to save data on the client side, and Cache-based methods using `E-Tags` [3].

2.1.2 Stateless tracking

Stateless tracking combines information about the target system via browser APIs and network information, which, when combined, creates a unique and persistent identifier for this device or browser [9, 25]. It differs from stateful methods in that this value is a product of the host system, rather than a saved state, and therefore cannot be deleted or cleared by the user.

Certain hardware attributes, which on their own may not be unique, when combined create a unique digital fingerprint, which renders it possible to identify a particular browser on a particular device [9]. This method will usually require code execution, either via JavaScript or Flash, which is enable gather the data from APIs which provide device attributes like the device resolution, browser window size, installed fonts and plugins, etc [25]. More advanced methods leverage observations of the ways different hardware render HTML Canvas data [5, 24] or manipulate audio data in order to generate fingerprints [11].

2.1.3 Measuring Tracking Methods

In most cases, both *stateful* and *stateless* tracking can be measured from the browser. Measurement of *stateful* tracking is made easier by the origin requirements of the APIs being used. Both Cookies and localStorage sandbox data according to the domain name used by the accessing resource. For example, if a cookie is set for the domain `track.example.com`, this cookie can only be sent for requests to this address. This necessitates that trackers using these methods must always use the same domain in order to track across different sites. Thus, this origin requirement enables us measure a particular tracker’s presence across the web via the presence of a particular third-party domain—the identifier cannot be read by other domains

Stateless tracking does not have the same origin constraints as stateful tracking, therefore fingerprints could be transmitted to different domains, and then aggregated on the server side. Even though the use of stateful tracking is easier, due to the prevalence of browsers which will accept third-party cookies, we find that most trackers still centralise their endpoints. This is true also when 3rd parties engage in stateless tracking.

As stateless tracking uses legitimate browser APIs, we cannot assume simply that the use of these API implies that tracking is occurring. We use a method, based on our previous work, of detecting the transmission of data values which are unique to individual users [30]. We detect on the client side which values are unique based on a k-anonymity constraint: values which have been seen by fewer than k other users are considered as *unsafe* with respect to privacy. We can use this method as a proxy to measure attempted transmission of fingerprints generated with stateless tracking, as well as attempts to transmit identifiers from stateful methods over different channels.

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

2.2 Browser Instrumentation

We measure tracking in the browser using a browser extension. This enables us to observe all requests leaving the browser and determining if they are in a tracking context or not. For each page loaded by the user, we are able to build a graph of the third-party requests made and collect metadata for each.

HTTP and HTTPS requests leaving a browser can be observed using the `webRequest` API [1]. This is a common API available on all major desktop web browsers. It provides hooks to listen to various stages of the lifecycle of a request, from `onBeforeRequest`, when the browser has initially created the intent to make a request, to `onCompleted`, once the entire request response has been received. These listeners receive metadata about the request at that point, including the url, resource type, tab from which the request originated, and request and response headers.

We first implement a system for aggregating information on a page load in the browser, enabling metadata, in the form of counters, to be added for each third-party domain contacted during the page load. We define a page load as being:

- Created with a web request of type `main_frame` in a tab;
- Containing the `hostname` and `path` extracted from the `URL` of the main frame request;
- Ending when another web request of type `main_frame` is observed for the same tab, or the tab is closed.

For each subsequent request for this tab, we assess whether the hostname in the url is third-party or not. This is done by comparing the Top-Level-Domain+1 (TLD+1)[3] forms of the page load hostname to that of the outgoing request. If they do not match, we add this domain as a third-party to the page load.

We collect metadata on third-party requests in three stages of the `webRequest` API: `onBeforeRequest`, `onBeforeSendHeaders`, `onHeadersReceived`.

In `onBeforeRequest` we first increment a counter to track the number of requests made for this domain. Additionally we count:

- the HTTP method of the request (GET or POST);
- if data is being carried in the url, for example in the query string or parameter string;
- the HTTP scheme (HTTP or HTTPS);
- whether the request comes from the main frame or a sub frame of the page;
- the content type of the request (as provided by the `webRequest` API);
- if any of the data in the url is a user identifier, according to the algorithm from [30];

[3]Top level domain plus first subdomain.

In `onBeforeSendHeaders` we are able to read information about the headers the browser will send with this request, and can therefore count whether cookies will be sent with this request.

In `onHeadersReceived` we see the response headers from the server. We count:

- that this handler was called, to be compared with the `onBeforeRequest` count;
- the response code returned by the server;
- the content-length of the response (aggregated for all seen third-party requests);
- whether the response was served by the browser cache or not;
- whether a `Set-Cookie` header was sent by the server;
- the origin country of the responding server (based on a geoip lookup of the IP address[4]).

As this code runs alongside Ghostery's blocking, we can also measure if requests were blocked by this extension. Depending on user configuration, this may be category related blocking, specific block rules, or based on Adblock blocklists.

Together, these signals give us a a high level overview of what third-parties are doing in each page load:

- Cookie's sent and `Set-Cookie` headers received (in a third-party context) can indicate *stateful* tracking via Cookies. Empirical evaluation shows that the use of non-tracking cookies by third-parties is limited.
- HTTP requests on HTTPS pages show third-parties causing mixed-content warnings, and potentially leaking private information over unencrypted channels.
- The context of requests (main or sub frames) indicate how much access to the main document is given to the third-party.
- The content types of requests can tell us if the third-party is permitted to load scripts, what type of content they are loading (e.g. images or videos), and if they are using tracking APIs such as beacons [29].
- The presence of user identifiers tells us that the third-party is transmitting fingerprints with requests, such as viewport sizes, or other tracking parameters.
- The difference between the number of requests seen by the `onBeforeRequest` and `onHeadersReceived` handlers indicates the presence of external blocking of this third-party, either at the network level or by another browser extension. We also measure if the extension hosting the measurement code blocked the request. This gives a measure of actual blocking due to Ghostery or Adblocker blocklists

[4]We use the MaxMind database for this purpose: `https://dev.maxmind.com/geoip/geoip2/geolite2/`

in the wild.

Once the described data on a page load has been collected, it is transmitted as a payload containing: the page's protocol (HTTP or HTTPS), the first-party hostname and path, and the set of third-parties on the page (*TP*).

$$pageload = \langle protocol, hostname, path, TP \rangle \quad (1)$$

The set of third-parties simply contain the third-party hostnames with their associated counters:

$$TP = \{\langle \texttt{hostname}, C \rangle, \ldots\} \quad (2)$$

The nature of this data already takes steps to avoid recording at a level of detail which could cause privacy side-effects. In Section 3 we will describe these steps, and the further steps we take before transmitting this data, and in the transmission phase to prevent any linkage between any page load messages, nor any personal information in any individual message.

3. PRIVACY-PRESERVING DATA COLLECTION

The described instrumentation collects information and metadata about pages loaded during users' normal web browsing activities. The collection of this information creates two main privacy challenges: First, an individual page load message could contain information to identify the individual who visited this page, compromising their privacy. Second, should it be possible to group together a subset of page load messages from an individual user, deanonymization becomes both easier, and of greater impact [28, 10]. In this section we discuss how these attacks could be exploited based-on the data we are collecting, and then, how we mitigate them.

3.1 Preventing message deanonymisation

The first attack attempts to find information in a *pageload* message which can be linked to an individual or otherwise leak private information. We can enumerate some possible attack vectors:

Attack 1. The first-party *hostname* may be private. Network routers or DNS servers can arbitrarily create new hostnames which may be used for private organisation pages. A page load with such as hostname may then identify an individual's network or organisation.

Attack 2. The *hostname path* combination often gives access to private information, for example sharing links from services such as Dropbox, Google Drive and others would give access to the same resources if collected. Similarly password reset urls could give access to user accounts.

Attack 3. *hostname* and *path* combinations which are access protected to specific individuals could leak their identity if collected. For example, the twitter analytics page `https://analytics.twitter.com/user/jack/home` can only be visited by the user with twitter handle `jack` [23].

Attack 4. Third-party hostnames may contain user identifying information. For example, if an API call is made containing a user identifier in the hostname, it could be exploited to discover more about the user. While this is bad practice, as the user identifier is then leaked even for HTTPS connections, we have observed this in the wild [20].

We mitigate attacks 1. and 2. by only transmitting a truncated MD5 hash[5] of the first-party *hostname* and *path* fields. By obfuscating the actual values of these fields we are still able to reason about popular websites and pages — the hashes of public pages can be looked up using a reverse dictionary attack — but private domains would be difficult to brute force, and private paths (e.g. password reset or document sharing links) are unfeasible. Therefore this treatment has desirable privacy properties, allowing us to still collect information about private pages without compromising their privacy and that of their users.

This treatment also mitigates some variants of attack 3., however for sites with a predictable url structure and public usernames (like in our twitter analytics example), it remains possible to lookup specific users by reconstructing their personal private url. We prevent this by further truncating the path before hashing to just the first level path, i.e. `/user/jack/home` would be truncated to `/user/` before hashing.

Attack 4. cannot be mitigated with the hashing technique, as we need to collect third-party domains in order to discover new trackers. We can, however, detect domains possibly using unique identifiers by counting the cardinality of subdomains for a particular domain, as well as checking that these domains persist over time. After manually checking that user identifiers are sent for this domain, we push a rule to clients which will remove the user identifier portion of these hostnames. We also report these cases to the service providers, as this practice represents a privacy leak to bad actors on the network. We can further reduce the probability of collecting unique subdomains by truncating all domains to TLD+2 level.

3.2 Preventing message linkage

Even if individual messages cannot be deanonymised, if messages can be linked it is possible that as a group they can be deanonymised, as shown in recent examples deanonymising public datasets [28, 10]. Furthermore, if an individual message happens to leak a small amount

[5]While using truncated hashes does not bring improved privacy properties, it does provide plausible deniability about values in the data.

of information, once linked with others the privacy compromise becomes much greater. Therefore, we aim to prevent any two *pageload* messages from being linkable to one-another.

The linking of messages requires the message sent from an individual user to be both unique, so that it does not intersect with others', and persistent, so that it can be used to link multiple messages together. We can enumerate some possible attacks:

Referring to attack 4 from the previous section may also be used for linkage, if the unique hostname is used over several popular sites. For example a case we found with Microsoft accounts was prevalent across all Microsoft's web properties when a user was logged in. The third-party domain was specific to their account and did not change over time. This third-party domain would therefore be used to link all visits to Microsoft sites indefinitely.

Attack 5. In a previous version of our browser instrumentation we collected the paths of third-party resources as truncated hashes. However, some resource paths could then be used for message linkage, for example avatars from third-party services such as Gravatar could be used to link visits on sites which display this avatar on every page for the logged in user. For this reason we removed collection of these paths.

Attack 6. Some third-party requests can be injected into pages by other entities between the web and the user. ISPs can intercept insecure web traffic, Anti-virus software often stands as a Man in the Middle to all connections from the browser, and browser extensions can also inject content in the page via Content scripts. Any of these entities can cause additional third-parties to appear on page loads. It is possible that a combination of injected third-parties could become unique enough to act as a fingerprint of the user which could link page loads together.

Attack 7. When data is uploaded from clients to our servers we could log the originating IP addresses of the senders in order to group the messages together, or utilise a stateful method to transmit user identifiers with the data.

We have already presented mitigations for the first two attacks. Attack 6. is difficult to mitigate for two reasons. Firstly, of the injected third-parties which we do detect, we cannot quantify the number of distinct users affected from the data that we collect. Therefore, it is not possible at the moment to calculate if certain combinations of third-parties would be able to uniquely identify an individual user. Secondly, a large proportion of these third-parties are injected by malware or other malicious actors, which implies an unstable ecosystem, where, as extensions get blocked and domains get seized, the set of injected third-parties will change. This also will have the effect that the persistence of the links will be limited. Despite this we aim to develop a mitigation method as part of our future work.

Attack 7 looks at the case where we ourselves might be either malicious or negligent as the data collector, creating a log which could be used to link the collected page loads back to pseudo-anonymous identifiers. It is important, that when monitoring trackers, we do not unintentionally become one ourselves. Trust is required, both that our client side code does not generate identifiers to be transmitted to the server along side the data, and that the server does not log IP addresses from which messages are received.

Trust in the client side is achieved by having the extension code open-sourced[6], and the extension store review and distribution processes should, in theory, prevent a malicious patch being pushed to diverge from the public code. Furthermore, extensions can be audited in the browser to allow independent inspection of requests leaving the browser.

In order to allow the client to trust that the server is not using network fingerprints to link messages, we have developed a system whereby data is transmitted via proxies that can be operated by independent entities. Encryption is employed such that these proxies cannot read or infer anything about that transmitted data. The scheme is therefore configured such that the data collection server only sees data messages—striped of user IPs—coming from the proxies. The proxies see user IP addresses and encrypted blobs of data. Proxies visibility of message transmissions is limited by load-balancing, which partitions the message space between the acting proxies, limiting how much metadata each is able to collect. The client-side part of this system also implements message delay and re-ordering to prevent timing-based correlations [22].

The deployment of this system means that, if the user trusts the client-side implementation of this protocol, and the independence of the proxies, then he does not have to trust our data collection server to be sure we are not able to link messages together.

3.3 Privacy Evaluation

We have evaluated the risks in collecting the data gathered through our described browser instrumentation, and several steps which we take to mitigate and prevent these risks from being exploitable. We cannot prove completely anonymized data collection - we have made several improvements in response to findings from both internal and independent external audits of this data - however we regard this methodology as being robust, and if the data were to be leaked we are confident that the privacy consequences would be minimal.

[6]https://github.com/cliqz-oss/browser-core

4. DATA AGGREGATION

In this section we describe how the collected page load messages are aggregated to provide high-level statistics which describe the tracking ecosystem.

In previous studies of the tracking ecosystem, third-party domains have been truncated to TLD+1 level, and then aggregated. The reach of, for example `google-analytics.com`, will be then reported as the number of sites which have this domain as a third-party. This is a simple and easily understandable aggregation method, however it has some shortcomings:

- A domain name is not always transparent. For example it will not be apparent to everyone that the domain `2mdn.net` is operated by Google's Doubleclick advertising network. It is important that the entities of the aggregation are meaningful and transparent.
- Domain level aggregation will duplicate information for service which use multiple domains in parallel. For example Facebook uses `facebook.net` to serve their tracking script, and then send tracking pixel requests to `facebook.com`, where the Facebook tracking cookie resides. According to domain semantics these are separately registered domains, though they will always occur together on web pages. Therefore reporting these two domains separately is redundant, and potentially misleading, as one might assume that the reach of the two entities can be added, when in fact they intersect almost entirely.
- Domain level aggregation will hide tracker entities who use a service on a subdomain owned by another organisation. The prime case here is Amazon's `cloudfront.com` CDN service. Several trackers simply use the randomly assigned `cloudfront.com` domains rather than use a CNAME to point to their own domain. For example New Relic[7] sometimes uses the `d1ros97qkrwjf5.cloudfront.net` domain. If we aggregate all Cloudfront domains together, the information about different trackers is lost.

We solve these issues by using a manually curated database, based on Ghostery's [12] tracker database, which maps domains and subdomains to the services and/or companies they are know to operate under, as a base. For a given domain, the database may contain multiple subdomains at different levels which are mapped to different services. When aggregating domains, we then find the matching $TLD + N$ domain in the database, with maximal N. i.e. if we have mappings for `a.example.com`, `b.example.com` and `example.com`, then `a.a.example.com` would match to `a.example.com`, while `c.example.com` would be caught by the catch-all `example.com` mapping. These mappings allow us to split and aggregate domains in order to best describe different tracking entities.

[7]New Relic is an performance analytics service which reaches over 4% of web traffic as measured by our data

4.1 Different measurements of reach

The page load data we collect allows us to measure tracker and companies' reach in different ways. We define a tracker or company's 'reach' as the proportion of the web in which they are included as a third-party. This is done by counting the number of distinct page loads where the tracker occurs:

$$reach = \frac{|page\ loads\ including\ tracker|}{|page\ loads|} \tag{3}$$

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.

$$site\ reach = \frac{|unique\ websites\ where\ tracker\ was\ seen|}{|unique\ websites|} \tag{4}$$

Differences between these metrics are instructive: *reach* is weighted implicitly by site popularity—a high *reach* combined with low *site reach* indicates a service which is primarily on popular sites, and is loaded a high proportion of the time on these sites. The inverse relation—low *reach* and high *site reach*—could be a tracker common on low traffic sites, or one which has the ability to be loaded on many sites (for example via high reach advertising networks), however does so rarely.

4.2 Aggregation of instrumentation counters

The reach metrics described are based on presence—when requests occur in a page to specific third parties. In Section 2.2 we described other counters we collect in order to measure use of potential tracking vectors. We aggregate these statistics by counting the number of pages where these methods are invoked at least once during the page load, then report this metric as the proportion of the tracker's *reach* which used this method. We report:

- Cookie tracking context – Cookies sent with request, or server responded with a `Set-Cookie` header.
- Fingerprinting context – User identifier detected in the request (as per [30]).
- Tracking context – Either cookie tracking or fingerprinting context, inclusive.
- Secure context – Only HTTPS requests for the page load.
- Content types – Pages where specific resource types were loaded by the tracker (e.g. scripts, iframes, plugins)
- Blocking effect – How often the tracker is affected by blocklist-based blockers.

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

5. RESULTS

Most studies analysing the tracking landscape have generally been performed in the context of one off measurements [11] or longitudinal surveys with limited scale and scope [17, 18]. In the remainder of this section, we look at these two perspectives: dissecting the tracking landscape data at a snapshot in time, and analysing longitudinal trends that reveal trends and could inform policy.

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.

It is important to note that unlike other studies, in which the measurement platform does not interact with websites in the same way real users would, [11], the data which will be subject to our analysis, has been generated by users of our browser extension over the course of the last year. As such, the behaviour of trackers and websites is what we see in reality.

The data spans from May 2017 to April 2018, amounting to a total number of page loads of 1.5 billion. This is the largest dataset on web tracking to our knowledge [11].

5.1 Snapshot in Time

We will be looking at the data from April 2018, composed of roughly 340 million page loads, and filtering the top 1330 most visited websites. We measure that 71% of the traffic to these sites contains tracking. The average number of trackers per site is 8, and the average number of tracking requests per page load 17.

5.1.1 First parties

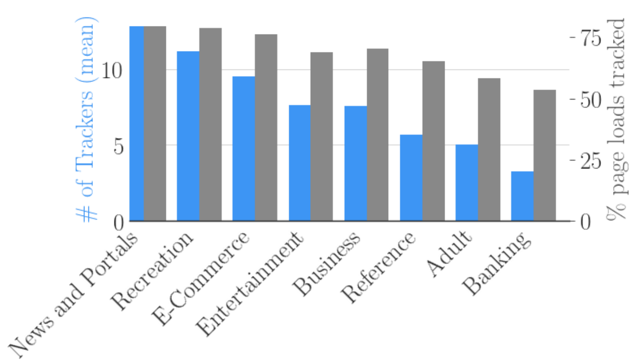

Figure 1: Tracking by website categories

In Figure 1 we see that websites in the category of News and Portals have the highest number of third parties at approximately 13 per page on average, with tracking occurring on 79 % of the measured page loads. Banking websites tend to have the lowest number of third parties as well as a lower percentage of page loads where tracking occurs.

5.1.2 The most prevalent third parties

Third parties often provide functionality that is not immediately distinguishable from or visible in the website they are present on. Hence, to achieve transparency and understand the tracking market structure, estimating the prevalence of a particular tracker defined in terms of the fraction of web traffic they are present on (reach), is important.

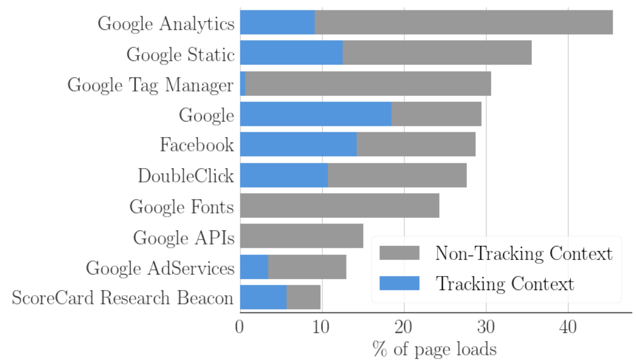

Figure 2: Top 10 third parties by reach

If we look at the top 10 third parties in Figure 2, we see that Google Analytics has the highest reach, being present on roughly 46% of the measured web traffic, and 8 out of the top 10 third parties are operated by Google.

Note that third parties do not always operate in a tracking context, which given our definition of third-party tracking, means they do not always send unique user identifiers. For instance, Google APIs is mostly used to load other 3rd parties such as Google Fonts and other static scripts, which is why we see it largely operating in a non-tracking context.

5.1.3 From trackers to organisations

By clustering third parties under parent organisations, we can also measure the reach of the latter.

We observe that third-party scripts owned by Google are present in about 82% of the measured web traffic, and operate in a tracking context for slightly less than half that time. Facebook and Amazon follow next, and generally the distribution of reach by organisation in Figure 3 has a long tail.

5.1.4 Third Parties: categories and consequences

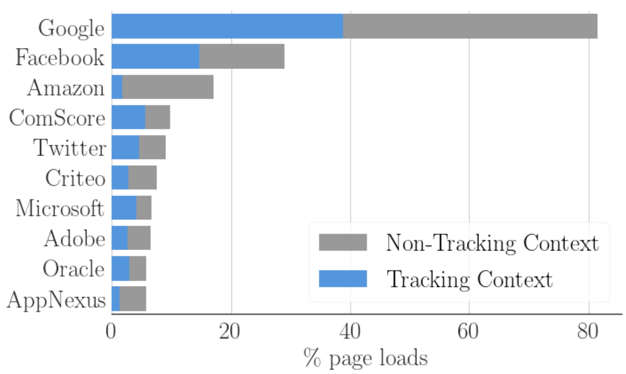

Figure 3: Top 10 organisations by reach

Most third parties are loaded to perform certain functionality that websites need. Note how among third parties with the highest reach in Figure 4, those that provide advertising services are predominant (left y-axis in blue), representing almost half of the trackers analysed in this study. In the same figure, we see the proportion of page loads containing a tracker of a given category was blocked by an ad-blocker.

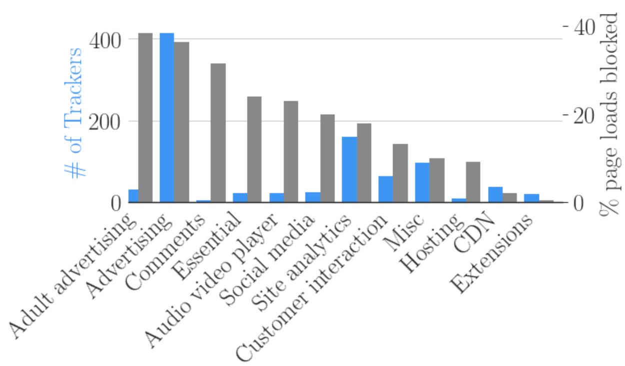

Figure 4: Third-party counts and block-rates by category

Note, that as our reach measurement occurs before blocking, these block rates are not reflected on the third-party and company reach we have already reported.

5.1.5 *Reach to Site Reach ratio*

Besides web traffic presence, we can also measure the first party presence for these third parties (site reach). The ratio of reach to site reach tells us an interesting story about the nature of the third party. The higher this ratio is, the more it suggests the third party being popular on few popular domains, and the lower it is, the more likely the third party could be some form of malicious software.

Take the example of DoublePimp with a a reach to site reach ratio of 28.8 (reach: 0.8% and site reach: 0.0002%), typically present on adult sites, and particularly in a few popular ones.

Domain	Method	Reach
kaspersky-labs.com	HTTP MITM	2.0%
worldnaturenet.xyz	Malicious extension	0.27%
eluxer.net	Malicious extension	0.20%
ciuvo.com	Price comparison ext	0.16%
comprigo.com	Price comparison ext	0.15%

Table 1: Man in the middle (MITM) trackers

Similarly, `eluxer.net`, with a reach to site reach ratio of 0.1, is a malicious extension which does insert tracking requests into pages as the user browsers.

5.1.6 *A new breed of tracker*

Our data also measures a previously unmeasured type of tracker - those placed not by website owners or ad networks, but by men in the middle. These are trackers which insert extra requests into pages either by intercepting network traffic on a device, or using browser extensions. The largest of these trackers is the anti-virus vendor Kaspersky, whose software installs new root certificates on the user's system in order to man-in-the-middle all requests from the operating system, and insert tracking requests into every HTML document. This method enables the tracking of 2% of total web browsing (i.e. participants with this software installed represent 2% of the collected page loads).

Table 1 shows the top 5 such trackers. From our investigations, `worldnaturenet.xyz` and `eluxer.net` both appear to be extensions installed via malware, which then track and inject advertising into pages. We were not able to determine the owners of these operations, but there are several others with similar characteristics in our data. In contrast, the `ciuvo.com` and `comprigo.com` browser extensions can be easily found, and the companies operating them.

5.1.7 *Regional Data flows*

In Section 2.2 we noted that we can observe the IP address of the responding server, and from that use a GeoIP database to retrieve the country this server is situated in. Using this data, we can assess data flows from users in specific countries to trackers located in others. Table 2 shows where third-party requests are loaded from for pages loaded from Australia, Germany, France, the UK, the Netherlands, Russia and the USA.

We can see that in most cases the majority of page loads are tracked by servers located in the USA. Tracking of US users rarely goes abroad - 7% of tracked pages make requests to Ireland - while in other regions US servers track on most pages. One exception is Russia, where Russian servers track marginally more pages than

From	% pages with 3rd party request to							
	AU	DE	FR	GB	IE	NL	RU	US
AU	26	1	0	0	5	1	2	**92**
DE	0	41	14	8	29	34	5	**79**
FR	0	11	31	7	21	19	4	**82**
GB	0	4	3	24	22	30	3	**81**
NL	0	7	4	4	29	38	4	**79**
RU	0	9	5	1	20	13	**64**	62
US	0	1	1	1	7	2	2	**98**

Table 2: Locations of third-party services accessed for users in different geographical regions.

Entity	All 3rd party requests secure		
	May 2017	April 2018	Change
Top sites	56.7%	81.1%	+24.4%
News sites	27.0%	68.0%	+41.0%
Google Analytics	64.0%	84.2%	+20.3%
Facebook	72.2%	83.9%	+11.7%
AppNexus	56.5%	83.5%	+27.0%
Snowplow	72.6%	46.3%	−26.4%

Table 3: HTTPS Adoption of sites and trackers from May 2017 to April 2018

those based in the USA (64% to 62%).

Note, a limitation of this result is the validity of GeoIP results from some data centres. Notably, Google IPs always resolve to be located in the USA with the database we use, despite servers actually being located worldwide.

5.2 Longitudinal

Longitudinal studies have typically been done on a smaller scale to one-off crawls [17, 18]. Having a clear snapshot view of tracking at scale is important, but this often means the dynamics of tracking over time, are lost.

In this section, we explore the data at different levels of granularity from measuring the data cost imposed on users by third parties to technology trends in the tracking landscape.

5.2.1 HTTPS Adoption

Previous studies have highlighted the issue of insecure third-party calls compromising the security and privacy of page loads [11]. In this work we measure the protocol of outgoing third-party requests from the browser. We can use this measurement to detect the adoption rates of HTTPS across sites, and specifically which trackers are lagging behind on this metric.

Table 3 shows how a selection of entities HTTPS usage has changed over the study period. We can see a strong increase in sites which have all third-party content loaded over HTTPS, from 57% to 81%. Certain categories of site lag behind in this regard though, namely news sites.

Looking at specific trackers, we can see dominant players such as Google Analytics and AppNexus successfully migrating clients to HTTPS over the year. Others, like Facebook, have had slower progress on this front.

In general, trackers improved their HTTPS usage over this period, with only 65 trackers (of 587 with data at both time points) not increasing HTTPS coverage. A small number of outliers showed a negative trend, for example Snowplow[8], an analytics provider present on major news websites, including Le Monde and New York Times.

5.2.2 Cost imposed on users

As users navigate the web, they load content from websites they visit as well as the third parties present on the website. On average, for each first party page load, there are 17 third-party tracking requests. So beyond the privacy erosion, there is a material cost involved in this transaction. 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 [13]. Here we measure the amount of data needed to load third-party content.

We take the sum of the `Content-Length` of all third-party requests in the top 1330 websites over the last year, and measure the variation in this data consumption overt time. The median content length per site from third parties was 0.42MB with an interquartile range (IQR) of 0.18-1.5MB, down from 0.58MB (IQR 0.24-1.8MB) a year earlier. The distribution has a long tail due to third parties offering Audio Video player services being part of the data.

5.2.3 Tracking technologies over time

There are several observations in how different content types are used in the context of tracking. The following are measured:

- script: Javascript code (via a `<script>` tag or web worker).
- iframe: A subdocument (via `<frame>` or `<iframe>` elements).
- beacon: Requests sent through the Beacon API.
- image: `Image` and `imageset` resources.
- 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.

[8] `https://snowplowanalytics.com/`

With this data we can see that, for example, during April 2018 Google Analytics loaded their script on each page load (97% of the time), then registered the visit via an image (pixel) on 50% of page loads. We also see that on 2.6% of pages a request is also made via the Beacon API.

In Figure 5 we see that scripts and images are the most popular content types for tracking. Interestingly, beacons, originally designed to satisfy tracking use cases are encountered increasingly less.

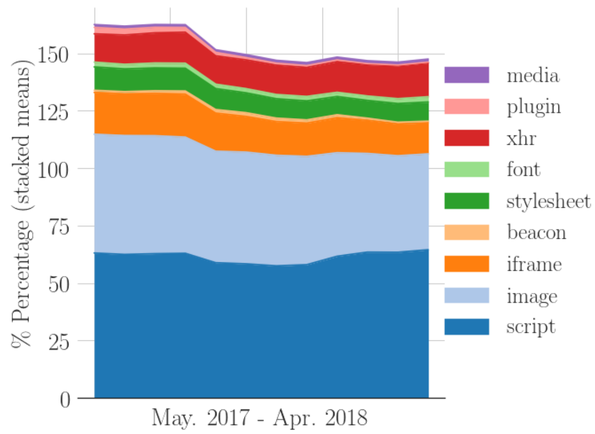

Figure 5: Content type usage for third parties

5.2.4 *Reach by type of third party over time*

The data also enables us to monitor the reach of third parties over time. If grouped and averaged as in Figure 6 we observe almost an across-the-board decrease in the reach of third parties, most notably in the category of extensions that engage in MITM tracking. One explanation could be attributed to an increased adoption of ad-blockers.

This analyses can be conducted at a more fine granular level, by monitoring the change in the average number of third parties in any given site. In Figure 7 we compare the average number of third parties present on The Guardian and Le Figaro with the industry average over the last year.

5.3 Discussion

Our results re-affirm previous findings: That significant numbers of third-parties are loaded on each page a user visits across the web. The number of third-parties is the highest on news websites, and where tracking is utilised. The number of trackers per page on a website generally trends with the presence of advertising networks and the Adtech supply chain which permits multiple parties to a bid to execute scripts on a page.

One surprising aspect may be the prevalence of tracking on business websites. This is again tied to Adtech

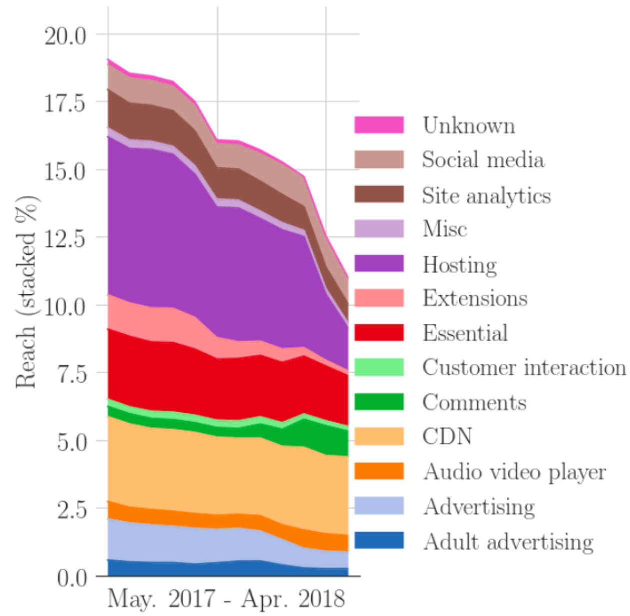

Figure 6: Reach over time by type of third party

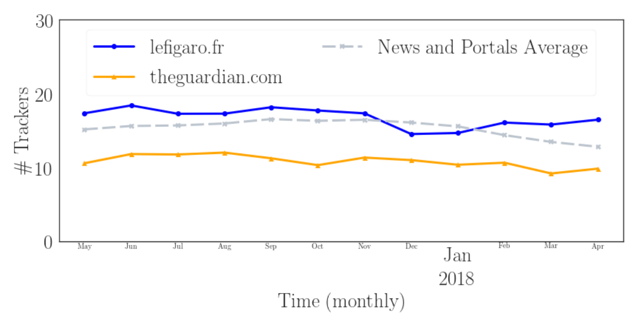

Figure 7: Third parties: The Guardian and Le Figaro

conversion measurement: business who advertise elsewhere on the web are encouraged to install their ad vendor's scripts on their pages in order to attribute landings to users who view particular ads. This enables fine-grained measurement of specific campaigns across ad platforms.

Figure 4 confirms that the largest category of third-parties is in advertising, but these are also the most heavily affected by blocking, with almost 40% of page loads seen by advertising trackers affected by blocking. This provides a extra layer of nuance of previous reports how the level of ad-blocker adoption [26], showing the amount of blocking within a population of ad-blocker users[9], taking whitelisting and gaps in blocklists into account.

[9]Ghostery and Cliqz both integrate an ad-blocker

Our longitudinal analyses show a decline in the number of third-parties loaded on pages. We may then infer that website owners are reducing the number of third-parties they allow to be loaded on their pages. It could also be tied to changes in Adtech triggered by the GDPR, where supply chains are being condensed in an attempt to become compliant, and to increase the chance of getting user consent for tracking [15]. However, one likely larger contributor to this drop is the aforementioned ad-blocking. As well as the blocking we measure from the resident browser extension, many users will have additional adblocking extensions or firewall rules installed to block certain third-parties. A side-effect of blocking ad networks, this a lower reported number of third-parties on the page, as by blocking the initial ad network request, subsequent vendors which would have been loaded by this first script are not seen. This has the effect of reducing the number of third-parties measured.

Of note, and concerning for websites trying to become compliant with data protection law, is our analysis of third-party content types. We measure that most of third-parties are permitted to load scripts into publishers' pages, and this is the most common way in which third-party content is embedded.

This is firstly a security issue - scripts loaded in the main document of a website have access to all page content, and can perform any action they wish. The prevalence of this practice makes malvertising—the serving of malware via advertising networks—possible, and presents a much larger attack surface against the site. In a recent when a third-party script was compromised and started loading a cryptocurrency mining script in the website of the Internet Commissioner's Office `ico.org.uk` in the UK and more than 4000 other websites where this third party was present [14].

Secondly, this is a compliance challenge. As scripts provide the third-parties with significant capabilities to ex-filtrate data from the website in which they are embedded, to be compliant website owners should require contracts to state the constraints under which the third-party must operate, such that any consent that the first-party obtains for data processing is valid for what the third-party actually does. Our position is that this is likely overly burdensome, and the adoption of privacy-by-design solutions would be preferable, where the system design enforces constraints on third-parties, and non-compliance is not technically possible.

A positive result of our longitudinal analysis is the continuing adoption of HTTPS by both first and third parties. A combination of nudges have encouraged providers to switch, making certificates easier to obtain via services such as LetsEncrypt[10], increased pressure from browser vendors, blocking some kinds of mixed-content and UI changes such as showing warnings on forms on insecure pages, and increased concerns about network eavesdroppers, such as ISPs. Progress, however, is still dependant on the third-party vendors used, as our results show. Some services have achieved better progress than others in this regard.

Note that our results for HTTPS adoption may over estimate in some aspects. A proportion of participants (those using the Cliqz browser) have the HTTPS Everywhere[11] installed and enabled by default, and this extension will prevent loading of insecure sites when a secure version is available, thus increasing the reported HTTPS adoption rate.

Our results also measure a new kind of tracking - that of browser extensions, malware and other software injecting requests into pages browsed by users. While the presence of spyware and malware in browser extension stores is not new, our results provide a first look at its prevalence in the wild. We hope that this data can be used by browser vendors to detect malicious extensions, or when users' privacy could be compromised by malware on their system.

6. WHOTRACKS.ME WEBSITE

One of the contributions is *WhoTracks.Me* , a website that hosts the largest dataset of tracking on the web and detailed analysis of the growing body of data, neatly organised around detailed profiles of first and third parties, respectively referred to as websites and trackers.

For each website, we provide a list of data that infers the tracking landscape in that website. The per site third-party data includes, but is not limited to: the number of third parties detected to be present at an average page load of that website as well as the total number of third parties observed in the last month; the frequency of appearance for each third party; the tracking mechanisms used by third parties in the site; a distribution of services the present third parties perform on that page Heavy use of data visualisations is made to make the data accessible to as wide a spectrum of an audience as possible.

Given the often obscure profiles of trackers, for each tracker we try to identify the organisation operating it and make the information accessible. For each tracker profile, we provide the information needed to identify them; the list of domains it uses to collect data; the organisation that operates them; reach and site reach as defined in equations 3 and 4), as well as the methods they use for tracking. Furthermore, we provide information on the distribution of the types of websites they are seen to be present, similar third parties and a list of sites where it has been seen to be present. For an example, please visit a tracker profile on *WhoTracks.Me* .

[10] `https://letsencrypt.org/`

[11] `https://www.eff.org/https-everywhere`

6.1 Who is WhoTracks.Me for?

WhoTracks.Me is a monitoring and transparency tool. We have open sourced data from more than 1.5 billion page loads per month, and plan to continue the effort. As tersely demonstrated in Section 5, the possibilities for using the data are numerous and the users diverse:

- **Researchers** - Can use the open data to investigate tracking technologies, develop more comprehensive protection mechanisms and threat models, investigate the underlying structure of online tracking as a marketplace etc.
- **Regulators** - The ability to access both detailed snapshots of tracking data as well as observe entities over time, enables regulators to use *WhoTracks.Me* as a monitoring tool to measure the effect of regulations like the General Data Protection Regulation (GDPR) [8] and ePrivacy [2].
- **Journalists** - Regardless of whether one takes the angle of the market structure of online tracking, or conceptually facilitating the education of the consumers on the issue, journalists will have enough data to derive insights from.
- **Web Developers** - Certain third-party scripts that web developers may add to their sites, have the capacity of loading other third parties, which the web developer may or may not know about. This, for instance, is the typical behaviour of ad networks like DoubleClick. Web developers can use *WhoTracks.Me* to keep an eye on the extent to which they retain control over third parties loaded, which will be important in the context of GDPR compliance [8]. Furthermore, not doing so can often have undesired consequences.
- **Block-list maintainers** - Can benefit from the automatic discovery of trackers, and can easily use the open source data to generate block lists[12].
- **Everyone** - Can build understanding of their exposure to tracking by learning about the tracking landscape on their favourite websites and read the educational resources in the *WhoTracks.Me* blog.

7. SUMMARY & CONCLUSIONS

As the line between the physical and online lives becomes more blurred, we believe online privacy will gain the attention of academics, regulators, media and users at large. In the context of paving the way for a constructive approach to dealing with online tracking, we open source the *WhoTracks.Me* data, which we plan to maintain, and update on a monthly basis.

This paper, and the living representation of it: *WhoTracks.Me* , contribute to the body of research, and public sphere more broadly, in the following ways:

- **Largest dataset on web tracking** to our knowledge. This assists researchers, regulators, journalists, web developers and users in developing efficient tools, devising policies and running awareness campaigns to address the negative externalities tracking introduces.
- **Longitudinal data**: While snapshots of data are necessary, in the non-transparent environment of online tracking, for the purpose of monitoring, it is also important to have have longitudinal data. *WhoTracks.Me* open sources data from the longest measurement of web tracking to date.
- **Measuring without platform-side-effects**: The data is generated by the behaviour of real users, which means the data is not prone to effects introduced by the measuring platform.
- **Human-Machine cooperation**: A significant amount of browser privacy tools, rely on publicly maintained block lists. *WhoTracks.Me* data contains trackers profiled algorithmically, as presented in [30]. Assisting the maintenance of blocklists, the community can focus on the accuracy of demographic data of the identified trackers, thus collectively improving transparency.
- **Measuring the effects of regulation**: The longitudinal nature of the data, enables users of *WhoTracks.Me* to measure the effects of regulation on the tracking landscape. An example of such application will be the measuring of effects the implementation of the General Data Protection Regulation (GDPR), in May 2018 will have on tracking practices.

Given increasing concern over the data collected by often nameless third-parties across the web, and consumers' struggles to keep control of their data trails, more transparency, accountability and monitoring is required in the ecosystem. This work represents a step-change in the quantity and depth of information available to those who wish to push for a healthier web.

8. REFERENCES

[1] chrome.webRequest. `https://developer.chrome.com/extensions/webRequest`. Accessed: 2018-05-25.

[2] ePrivacy Directive. `http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=CELEX:32002L0058:en:HTML`. Accessed: 2018-02-04.

[3] MDN Web Docs: E-Tag. `https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/ETag`. Accessed: 2018-03-02.

[4] MDN Web Docs: LocalStorage. `https://developer.mozilla.org/en-US/docs/Web/API/Window/localStorage`. Accessed: 2018-03-02.

[5] G. Acar, C. Eubank, S. Englehardt, M. Juárez,

[12] `https://whotracks.me/blog/generating_adblocker_filters.html`

A. Narayanan, and C. Díaz. The web never forgets: Persistent tracking mechanisms in the wild. In G. Ahn, M. Yung, and N. Li, editors, *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security, Scottsdale, AZ, USA, November 3-7, 2014*, pages 674–689. ACM, 2014.

[6] A. Cahn, S. Alfeld, P. Barford, and S. Muthukrishnan. An empirical study of web cookies. In J. Bourdeau, J. Hendler, R. Nkambou, I. Horrocks, and B. Y. Zhao, editors, *Proceedings of the 25th International Conference on World Wide Web, WWW 2016, Montreal, Canada, April 11 - 15, 2016*, pages 891–901. ACM, 2016.

[7] J. M. Carrascosa, J. Mikians, R. C. Rumín, V. Erramilli, and N. Laoutaris. I always feel like somebody's watching me: *measuring online behavioural advertising*. In F. Huici and G. Bianchi, editors, *Proceedings of the 11th ACM Conference on Emerging Networking Experiments and Technologies, CoNEXT 2015, Heidelberg, Germany, December 1-4, 2015*, pages 13:1–13:13. ACM, 2015.

[8] Council of the European Union and European Parliament. Regulation (eu) 2016/679 of the european parliament and of the council of 27 april 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing directive 95/46. *Official Journal of the European Union (OJ)*, 59:1–88, 2016.

[9] P. Eckersley. How unique is your web browser? In M. J. Atallah and N. J. Hopper, editors, *Privacy Enhancing Technologies, 10th International Symposium, PETS 2010, Berlin, Germany, July 21-23, 2010. Proceedings*, volume 6205 of *Lecture Notes in Computer Science*, pages 1–18. Springer, 2010.

[10] S. Eckert and A. Dewes. Build your own nsa. In *33rd Chao Computer Club Congress, 2016, Hamburg, Germany*, Dec 2016.

[11] S. Englehardt and A. Narayanan. Online tracking: A 1-million-site measurement and analysis. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, Vienna, Austria, October 24-28, 2016*, pages 1388–1401, 2016.

[12] Ghostery. Ghostery. https://ghostery.com/".

[13] M. Hanson, P. Lawler, and S. Macbeth. The tracker tax: the impact of third-party trackers on website speed in the united states. Technical report, 2018. Available at: `https://www.ghostery.com/wp-content/themes/ghostery/images/campaigns/tracker-tax/Ghostery_Study_-_The_Tracker_Tax.pdf`.

[14] S. Helme. Protect your site from cryptojacking with csp + sri. `https://scotthelme.co.uk/protect-site-from-cryptojacking-csp-sri/`. Accessed: 2018-05-25.

[15] J. Hercher. Google's gdpr consent tool will limit publishers to 12 ad tech vendors. `https://adexchanger.com/online-advertising/googles-gdpr-consent-tool-will-limit-publishers-to-1` Accessed: 2018-05-25.

[16] A. J. Kaizer and M. Gupta. Characterizing website behaviors across logged-in and not-logged-in users. In P. Gill, J. S. Heidemann, J. W. Byers, and R. Govindan, editors, *Proceedings of the 2016 ACM on Internet Measurement Conference, IMC 2016, Santa Monica, CA, USA, November 14-16, 2016*, pages 111–117. ACM, 2016.

[17] B. Krishnamurthy and C. E. Wills. Privacy diffusion on the web: a longitudinal perspective. In J. Quemada, G. León, Y. S. Maarek, and W. Nejdl, editors, *Proceedings of the 18th International Conference on World Wide Web, WWW 2009, Madrid, Spain, April 20-24, 2009*, pages 541–550. ACM, 2009.

[18] A. Lerner, A. K. Simpson, T. Kohno, and F. Roesner. Internet jones and the raiders of the lost trackers: An archaeological study of web tracking from 1996 to 2016. In T. Holz and S. Savage, editors, *25th USENIX Security Symposium, USENIX Security 16, Austin, TX, USA, August 10-12, 2016.* USENIX Association, 2016.

[19] S. Macbeth. Tracking the trackers: Analysing the global tracking landscape with ghostrank. Technical report, Ghostery, December 2017. Available at: `https://www.ghostery.com/wp-content/themes/ghostery/images/campaigns/tracker-study/Ghostery_Study_-_Tracking_the_Trackers.pdf`.

[20] S. Macbeth. Cliqz researchers discover privacy issue on bing.com and other microsoft sites, 2018. `https://cliqz.com/en/magazine/cliqz-researchers-discover-privacy-issue-bing-com-mi`

[21] G. Merzdovnik, M. Huber, D. Buhov, N. Nikiforakis, S. Neuner, M. Schmiedecker, and E. R. Weippl. Block me if you can: A large-scale study of tracker-blocking tools. In *2017 IEEE European Symposium on Security and Privacy, EuroS&P 2017, Paris, France, April 26-28, 2017*, pages 319–333. IEEE, 2017.

[22] K. Modi, A. Catarineu, P. Claßen, and J. M. Pujol. Human web overview. Technical report, Cliqz, 2016. Available at: `https://gist.github.com/solso/423a1104a9e3c1e3b8d7c9ca14e885e5`.

[23] K. Modi and J. M. Pujol. Data collection without privacy side-effects. Technical report, Cliqz, 2017. Available at: `http://josepmpujol.net/public/papers/big_green_tracker.pdf`.

[24] K. Mowery and H. Shacham. Pixel perfect: Fingerprinting canvas in html5. In *Proceedings of W2SP 2012*. IEEE Computer Society, 2012.

[25] N. Nikiforakis, A. Kapravelos, W. Joosen, C. Kruegel, F. Piessens, and G. Vigna. Cookieless monster: Exploring the ecosystem of web-based device fingerprinting. In *2013 IEEE Symposium on Security and Privacy, SP 2013, Berkeley, CA, USA, May 19-22, 2013*, pages 541–555. IEEE Computer Society, 2013.

[26] PageFair. The state of the blocked web: 2017 global adblock report. `https://pagefair.com/downloads/2017/01/PageFair-2017-Adblock-Report.pdf`. Accessed: 2018-05-25.

[27] P. Papadopoulos, P. R. Rodríguez, N. Kourtellis, and N. Laoutaris. If you are not paying for it, you are the product: how much do advertisers pay to reach you? In S. Uhlig and O. Maennel, editors, *Proceedings of the 2017 Internet Measurement Conference, IMC 2017, London, United Kingdom, November 1-3, 2017*, pages 142–156. ACM, 2017.

[28] J. Su, A. Shukla, S. Goel, and A. Narayanan. De-anonymizing web browsing data with social networks. In R. Barrett, R. Cummings, E. Agichtein, and E. Gabrilovich, editors, *Proceedings of the 26th International Conference on World Wide Web, WWW 2017, Perth, Australia, April 3-7, 2017*, pages 1261–1269. ACM, 2017.

[29] W3C. Beacon. `https://w3c.github.io/beacon/`. Accessed: 2018-05-25.

[30] Z. Yu, S. Macbeth, K. Modi, and J. M. Pujol. Tracking the trackers. In *Proceedings of the 25th International Conference on World Wide Web, WWW 2016, Montreal, Canada, April 11 - 15, 2016*, pages 121–132, 2016.