

C O V E N T R Y
U N I V E R S I T Y

Faculty of Engineering, Environment and Computing

School of Computing, Electronics and Mathematics

MSc Cyber Security

7030CEM - Cyber Security Individual Project

**Inspection of Web Tracker functioning methodologies,
techniques, and measures to minimize online exposure**

Author: Vishal Pratap Rayan

SID: 10616129

Supervisor: Dr. James Shuttleworth

Submitted in partial fulfilment of the requirements for the Degree of Master of Science in MSc Cyber
Security

Academic Year: 2020/21

Declaration of Originality

I declare that this project is all my own work and has not been copied in part or in whole from any other source except where duly acknowledged. As such, all use of previously published work (from books, journals, magazines, internet etc.) has been acknowledged by citation within the main report to an item in the References or Bibliography lists. I also agree that an electronic copy of this project may be stored and used for the purposes of plagiarism prevention and detection.

Statement of Copyright

I acknowledge that the copyright of this project report, and any product developed as part of the project, belong to Coventry University. Support, including funding, is available to commercialise products and services developed by staff and students. Any revenue that is generated is split with the inventor/s of the product or service. For further information please see www.coventry.ac.uk/ipr or contact ipr@coventry.ac.uk.

Statement of Ethical Engagement

I declare that a proposal for this project has been submitted to the Coventry University ethics monitoring website (<https://ethics.coventry.ac.uk/>) and that the application number is listed below.

Signed:



Date: 16-Jun-2021

First Name:	Vishal
Last Name:	Pratap Rayan
Student ID number	10616129
Ethics Application Number	P122719
1 st Supervisor Name	Mr. James Shuttleworth
2 nd Supervisor Name	Mr. Yasir Khan

Abstract

Keywords: Online privacy, Third-party tracking, FLOC, web trackers.

The study inspects the most used desktop web tracking techniques. By examining the underlying working principles and analysing the current state of those trackers. Automated crawls are performed on UK Top 50 domains to gain a better understanding of distribution of third party trackers and fingerprinting scripts. The second part of the report consists of testing different privacy enhancing tools such as uBlock origin, NoScript, Adblock plus etc. to compare the crawl results after installation. From gathered results, three sets of configurations are finally recommended.

Contents

Abstract	2
Acknowledgements	5
1 Introduction	6
1.1 Project Objectives	7
1.1.1 Scope	7
1.2 Overview of This Report	7
Part - I	8
2 Literature Review	8
2.1 FLoC	8
2.2 Third-Party Tracking	8
2.3 Evercookies	9
2.4 Fingerprinting	9
2.5 Privacy Enhancement	10
3 Methodology	11
3.1 Web Trackers	11
3.1.1 Types of Web Trackers	11
4 Analysis of Current Tracking Technology	25
4.1 Test Environment	25
4.1.1 Auditing tools	25
4.2 Analysis: Cookies	26
4.3 Analysis: Fingerprinting	28
4.4 Analysis: FLoC	29
Part - II	30
5 Preventive Measures: Testing	30
5.1 Browser	30
5.2 Browser configuration	31
5.2.1 Execution Blockers	32
5.2.2 Third-Party Tracker Blockers	33
5.3 Browsing behaviour	35
5.3.1 Cookie Consent	35
5.3.2 Search Engine Preference	36
5.3.3 Proxy	37
5.3.4 Browser Profile: Compartmentalization	37
5.4 Additional Tools	38
5.4.1 Virtual Private Network (VPN)	38
5.5 General good practices	38
5.5.1 FLoC Opt-out	38
5.5.2 Browser Storage	39
5.5.3 Public Email	39

6	Privacy Enhancement: Proposed Solution	40
7	Performance Comparison	42
8	Conclusion.....	43
	8.1 Achievements	43
	8.2 Future Work.....	43
9	Critical Appraisal.....	44
	Bibliography and References	45
	Appendix A – Requirements Specification Document	1
10	Virtual Machine specifications.....	1
	Ubuntu 64-bit.....	1
	Appendix B – Project Presentation	1
	Appendix C – Certificate of Ethics Approval.....	1
	Appendix X.....	2
	Abbreviations	2
	AudioContext Supported Browsers.....	3
	Usage Share of Desktop Browsers.....	3
	Privacy Footprint	4
	AudioContext properties:	5
	Fingerprint using DynamicsCompressor (sum of buffer values):.....	5
	Fingerprint using DynamicsCompressor (hash of full buffer):	5
	Fingerprint using OscillatorNode:	5
	Fingerprint using hybrid of OscillatorNode/DynamicsCompressor method:	5
	Crawl list	6
	Crawl Reports – Github links	7

Acknowledgements

I would like to thank my professor Dr. James Shuttleworth for guiding me with his valuable feedback. I would also like to thank my parents for their unconditional love and support which helped me pursue my masters. Lastly, I want to thank my dear friend Daniel with whom I had valuable discussions that shed light on topics that helped me with my project.

1 Introduction

The internet has evolved significantly over time as mass consumption has increased. Web pages and online services now offer various functionalities to enhance individual experience of users. The personalization of user experience is made possible by collecting user data at some point of time. This collection of personal data in many aspects has improved user experience. It is now common for websites to utilize external JavaScript files, CSS files, or analytics code from third-party providers. This is usually done as the first-party website the user intends to access may want to extend functionalities for themselves, as well as their users by incorporating third-party code. Information collected is often data about user browsing history, search history, links opened, products purchased, time and date of activity etc. The primary motivation of this online tracking as claimed by service providers, is to display personalized advertisements to the user. Displaying users advertisements for products they may be interested in increases the likelihood of purchase thus benefitting all involved parties.

This online tracking is achieved with the help of various web trackers. There are numerous tracking techniques used by service providers to successfully track user activity online. These can be classified into two main categories: (i) Stateful technologies, and (ii) Stateless technologies.

Data collection for personalised advertisements may seem harmless. However, it has been observed that collected data could be used for other purposes such as:

- a. *Third-degree price discrimination*: Where prices offered to consumers vary significantly based on their individual user profile and purchasing habits. Resulting in some consumers purchasing goods or services for prices higher than they should.
- b. *Estimate financial credit score*: Users' financial credit score have been assessed based on their social media contacts. Lenddo, a SaaS company assess user financial stability based on their Facebook friends and user scores have been observed to be negatively impacted based on their contact frequency with someone who was late to their loan payment (Lobosco, 2013).
- c. *Biased search results*: Search engines display users considerably different search results for the same search phrase based on their personal data. Reordering search results to possibly convince users to click on links search service providers could be incentivized for (Anderton, 2020).
- d. *Surveillance*: Government sponsored spy-agencies have shown to take advantage of mass online third-party tracking. By collecting huge amounts of global internet traffic and recording metadata including unique tracking IDs, even multiple devices sharing internet connection can be uniquely identified. Mass third-party tracking has facilitated this process by not needing to tap into device location sensors to obtain geolocation. Device IP address gives a rough estimate of the device location and to increase accuracy, access to device sensors is not necessary. Advertisement analytics software obtain this information and send back precise location details. United States' NSA program code-named 'HAPPYFOOT', exploits this to obtain accurate location details (Mayer, 2013).
- e. *Insurance*: Insurance companies desire to take a conservative approach with who they offer insurance to avoid bad candidates. Companies like Aviva, AIG and Prudential have been interested in data profiling to filter out user profiles that they consider risky (Bridges, 2011) . Based on their online activities, users are categorized into safe and risky categories.
- f. *Background check*: Employers use background check services like Checkr, Accurate, Certn, Xref etc. that evaluates a users' history to ensure they have no criminal records. This is good hiring practice but only if done right. The basis of a background check, if done purely online, could yield inaccurate results. Example, Potential employee

Kathleen Casey lost her job due to wrong background check assessment that concluded another person with the same name to be fraudulent. When Kathleen had committed nothing wrong (Robertson, 2011). This may not be a direct breach of privacy but the use of software to assess a user's online profile based on publicly available information opens up doors to sabotage attacks. When users are unable to protect their own online information, false information generated could affect one's life.

Internet privacy can often be assumed to what is being done online. While that may not be incorrect, a more accurate definition would be in identifying who the user is and what they are doing online. Data that could be used to identify users are commonly known as PII, Personally Identifiable Information. Users, often unknowingly, provide PII by merely visiting a website.

There now exists 'Data Broker' agencies that collect and sell such data legally. Every individual has the right to privacy and has data that needs to be confidential. Apart from users voluntarily providing information, data is also extracted in various other forms. This paper investigates the current literature on various tracking methods and techniques implemented by stateful tracking technologies and aims to propose a set of possible measures that could be adopted by the general user to limit their online footprint. The top English websites ranked on traffic by Alexa.com are identified and studied for the above-mentioned practices of processing user-data.

1.1 Project Objectives

This report was written to accomplish two main objectives. They are:

1. To educate reader about the massively performed online tracking that affects their everyday internet usage.
2. To provide reader a tested set of measures that could be adopted to limit their online exposure and improve privacy.

In this age of information, access to data is easier than ever before. It is, therefore, crucial to protect personal data. With increasing privacy-invasive tracking, regular internet users are at high risk of exposure of sensitive data.

1.1.1 Scope

Web tracking methods have diversified and there now exists multiple different ways to track users. The scope of this report will be limited to the most widely used online web tracking only. Specifically, browser-based web tracking.

1.2 Overview of This Report

This report is presented in two parts. Part I begins with the literature review which investigates the widely used web trackers, their functions, techniques, and methodology used to track users across the internet. Part I also shows emerging tracking technology where FLoC is discussed and analysed. An attempt to deeply understand web tracking is made to efficiently tackle this challenge.

In Part II, the importance of privacy is further explained by showing possible threats that could arise by excessive tracking and presents practical measures general users could take to limit their online footprint thus protecting their internet privacy. Comparison of various tools is carried out to pick the best set of tools. Part II is concluded by suggesting a guideline users could adopt to limit tracking by auditing performance of privacy enhancing tools and techniques.

Part - I

2 Literature Review

2.1 FLoC

Google's Federated Learning of Cohorts technology was still in its infancy while writing this report. Therefore, access to credible studies were limited. Majority of the research regarding FLoC came from reviewing the deeper functioning of the technology through its officially released whitepaper. Google released this whitepaper as part of its privacy sandbox initiative. Google has only recently cared to give importance to user privacy. Although they provide their users with delusional confidence of enhanced privacy, the overall Google ecosystem and their tremendous reach over the internet allows them to obtain more information about their users than they should. By simply observing the most visited websites and websites that use Google Analytics it can be inferred that the majority of Internet traffic at some point uses Google services. Therefore, Google's this attempt at privacy enhancement was immediately examined and heavily criticized by other privacy focused companies like Mozilla and DuckDuckGo. This criticism of FLoC by privacy pioneers sparked heated debates on online platforms. Researchers from Mozilla were one of the first to contribute to this work. Eric Rescorla and Martin Thomson from Mozilla discuss the potential flaws regarding privacy in FLoC. In their paper, 'Technical Comments on FLoC Privacy' they go over the technical aspects of FLoC where they attempt to discuss the threat model of FLoC. Which, as stated by them, is fairly challenging as FLoC is intentionally designed to leak browsing history. They observe how this data leakage is done to direct and indirect observers. Furthermore, they mention the level of fingerprinting that can be performed by FLoC. The report is finally concluded with some potential improvement measures.

2.2 Third-Party Tracking

Third party web tracking is often the heavily criticised component of web tracking. Researchers Jonathan R. Mayer and John C. Mitchell from Stanford University enlightened important key concepts on third party tracking with their paper 'Third-party Web Tracking: Policy and Technology' (Mayer & Mitchell, 2012) . In this paper, they explain the current state of web trackers by attempting to measure the web. Due to the lack of an automated tool that could monitor network traffic, automatically inspect browser state, and provide features to specify a custom measurement task, they developed FourthParty. FourthParty was created as a Firefox extension which is no longer supported but at the time it provided researchers with a good benchmark tool to carry their experiments. With the use of this web measurement tool the authors gain an objective, reliable score that they use to establish a basis for policymaking. FourthParty offers logging to a linked SQL database. Web measurement also provides them the ability to perform fast and automated scans to improve the quality of their results. They highlight the main problems regarding third party tracking such as easy availability of information including personally identifiable information online. The authors also mention the business models and trends regarding third-party tracking that gives readers a good idea about the processing of collected data. They were one of the early researchers to talk about the use of third-party tracking in cases apart from advertisement companies. The authors briefly discuss the underlying tracking technologies but do not do so in great detail. A 2017 research paper about the Web tracking mechanisms and implications by Bujlow (Bujlow et al., 2017) talks about all web tracking technologies at the time in great detail. This paper was exceptional work from the authors uncovering important topics. Since it was a 2017 study, they do not mention newer techniques like battery status API fingerprinting and FLoC simply because these technologies did not exist at that time. In 2013 paper 'Network Analysis of Third Party Tracking' Richard

Gomer et. al. (Gomer et al., 2013) present a technical report on the networking aspect of search providers that promote third party tracking. By showing a consistent network structure across search markets, they reveal a dominant connection across these networks. According to the authors, there is a 99.5% chance a user will fall victim to tracking by any of the top 10 trackers within 30 clicks of using search results. This study helps us understand the level of reach tracking companies have by comparing the tracking network structure to a small-world network to show how easily users can be tracked largely because a small group of entities are deeply connected. They also show the unpredictable distribution of third party trackers across search results that have no dependence on users' search query. The important takeaway from this paper is even with conscious, careful web usage by users it is almost inevitable to not encounter some form of third party tracking.

2.3 Evercookies

In 'The Web Never Forgets: Persistent Tracking Mechanisms in the Wild' authors Gunes Acar et. al (Acar et al., 2014) conducted one of the most comprehensive studies on persistent web tracking. The report uncovers obscure tracking techniques like Evercookies, Canvas fingerprinting, and introduces the reader to key persistent tracking techniques like Cookie syncing. Acar et. al performed the first automated study on Evercookies by showing the respawning of Cookies. They were also a part of the early introducers to Evercookie vector 'IndexedDB'. The authors emphasize on the likelihood of how easily even privacy-aware users can make mistakes. A single lapse in judgement could invalidate any privacy measure taken by user. It helps us understand that achieving optimal privacy is a process and even tech-savvy users struggle to maintain it. The paper describes the working with the data flow of IDs. Especially, the process of reviving deleted cookies. At the time of publishing the paper, since Evercookies were not a concept widely known, it had led to a lawsuit and a settlement worth \$500,000. Yet, due to the efficiency of tracking it provides it saw no decrease in implementation. In the paper, authors attempt to develop a reliable method to detect Canvas fingerprinting. This test was automated using a modified Firefox with Selenium browser automation. They performed a crawl of around 100,000 websites to gain more reliable information. The authors then carry out an experiment to detect user IDs by performing crawls on multiple machines to detect any identifying elements. This led to their discovery of Evercookies and Cookie respawning. Using the same method, when crawls were performed on multiple domains, some IDs were observed in multiple domains. Leading them to conclude that multiple domains could recognize the same user across the internet. Even if a user cleared his cookies and restarted browsing, trackers could place and sync a new set of user IDs to construct a new profile to begin tracking. The paper presents a few mitigation steps but also mentions how most users could fail to maintain privacy online.

2.4 Fingerprinting

Fingerprinting techniques have now become increasingly diversified with so many different methods that are brought to notice by Károly Boda et. al. in their 2012 paper (Boda et al., 2012). In this paper, Boda et. al. perform fingerprinting tests independent of browser. The aim of the authors was to obtain a unique fingerprint which was browser, plug-in, and domain independent. Along with other common fingerprinting interests such as installed fonts, and system features such as screen resolution and operating system, to achieve a reliable cross-domain fingerprinting method, the authors take carefully observe the first two octets of the user IP address. Which has shown to be constant in many cases, even when the user has a dynamic IP address that changes often. The authors also mention all the variable factors that could affect the result. More importantly, it is one of the few papers that talk about the factors that could

affect fingerprinting when performed on a large scale. Boda et al. perform the fingerprinting technique by collecting user data for six months from September 2010. In these months, they were able to perform a total of 989 tests from 615 different IP addresses. The test generated 662 user IDs. Majority of their research is directed towards proving the ability to uniquely identify browsers on the internet. The accuracy of the current state of fingerprinting methods are tested by using TOR browser and configuring proxy. Since, these browsers mainly obfuscate device IP address, other fingerprinting techniques are still valid and still successfully identify user. The observations made by comparing fingerprints generated through TOR browser to their initially generated dataset shows that IP spoofing done at client-side in hopes of hiding location data was also not sufficient. Trackers are still able to obtain system time zone. Which the general user is often unbothered to change. Boda et. al also bring to notice the inadequacy of simply using privacy-mode most browsers offer.

2.5 Privacy Enhancement

Privacy enhancement techniques concerning online browsing mainly involve browser choice, browser configurations and additional tools such as browser extensions. In 2019 paper, Johan Mazel et. al present a detailed study of the various widely-used browser extensions (Mazel et.al, 2019). The authors develop a systematic approach of testing the extensions that involve applying Kolmogorov Smirnov test to assess browsing metrics on privacy. The authors do not limit their research just to privacy protection but also take into consideration of how webpage quality could be affected from the extensions. They conclude the paper by producing test results suggesting NoScript extension to be the best in terms of minimising fingerprint. However, they also mention NoScript to negatively impact browsing experience. The results of this study combined with my own experiments discussed later in this report, help compile a tested list of extensions that are tested and recommended at the end of this report. In 2010 Gaurav Aggarwal et. al examine the privacy of private modes offered in browsers. This paper is focused more on local attackers but highlight some key points that also help in protecting privacy from web trackers. Results from this paper help us understand that private modes are not as efficient in protecting privacy as marketed and one must not solely rely on this feature to improve privacy.

3 Methodology

3.1 Web Trackers

Web trackers are code that is either embedded onto the website itself (First Party) or, referenced from an external source (Third party) to monitor the user's activities on their site. Data obtained from user activities are recorded allowing themselves to be tracked across the internet. This is a widely carried out practice which originally started with a simple log-based data collection solution that recorded basic website analytics such as number of site visits, pages visited and duration of visit. One such use case of Web trackers were 'Hit Counters'. These were used to record new IP addresses visiting the site for the first time to measure audience reach and engagement. Web trackers have evolved significantly since then and are now able to collect much more information from users. The switch from a simple log-based data collection to an advanced analytics tool was made possible using JavaScript tag-based analytics. As basic log records could not offer metrics to suit companies' growing needs. This switch provided interested businesses insights about user behaviours and growing trends. In 2005, Google released its web analytics tool named 'Google Analytics' which allowed any website to use their snippet to track their users on their site.

There are different kinds of web trackers available which are typically used in combination to get the most, reliable data out of users. Ranging from 'Forced tracking' such as pop-up windows to more subtle tracking techniques like HTTP Cookies, third-party tracking has seen a four-fold increase from 1996 to 2016 (Urton, 2016). Major tracking techniques are discussed in section 3.1.1.

3.1.1 Types of Web Trackers

There are stateful tracking technologies where browser state is preserved e.g., cookies and stateless technologies browser state is insignificant e.g., fingerprinting. All web trackers will fall into either one of those categories. Web trackers can be further classified into three categories:

- i. Storage-based tracking
- ii. Cache-based tracking
- iii. Fingerprinting tracking

Storage-based and Fingerprinting tracking are the most widely used tracking techniques.

3.1.1.1 Storage-based techniques

Storage-based tracking depends on storing data on the user's device. This data is referenced when needed to identify user.

3.1.1.1.1 HTTP Cookies

HTTP cookies are unique strings of data stored locally on user's device browser. In a client-server relationship, strings are generated and stored on the client's storage. This is achieved by sending response headers. HTTP requests consist of three key parts. A request line, headers, and the request entity. HTTP request line is where the method is specified. Request methods are shown in Table 1.

HTTP Request Methods							
GET	HEAD	POST	PUT	DELETE	CONNECT	OPTIONS	TRACE

Table 1 HTTP Request Methods

HTTP request headers contain the required meta-data for transfer of the cookie. Example of a HTTP request where request-line and request-headers are present is shown in Fig.1.

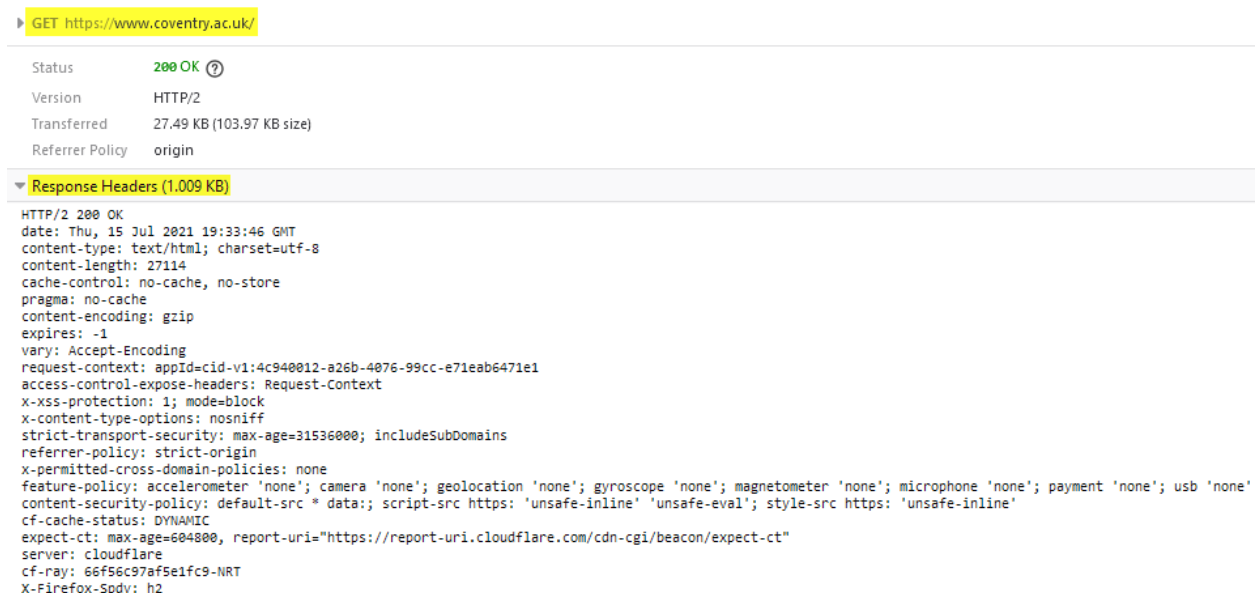


Figure 1 HTTP Request

GET requests do not contain any entity, unlike POST.

Server provides some arbitrary information to cookie with the Set-Cookie header. This information supplied could be something that the server would want to remember about the client. Depending on the use case, it could be data such as user ID, shopping cart items, buttons clicked, pages visited etc. allowing server to maintain a stateful interaction. Fig.2 shows server assigning SID value to the cookie.

```

== Server -> User Agent ==

Set-Cookie: SID=31d4d96e407aad42

== User Agent -> Server ==

Cookie: SID=31d4d96e407aad42

```

Figure 2 Set-Cookie example

This principal also applies to authentication cookies. User login information is remembered by the server not having to request that information for every page reload. For this reason, authentication cookie information is usually encrypted. Lifespan of a cookie could vary depending on its use case. While 'Session cookies' live only till the browser is closed, 'Persistent cookies' may exist much longer, usually, until a specific time and date. This means until its expiry which is set by its creator, persistent cookies will continue providing information to the server every time the site is visited.

```

== Server -> User Agent ==

Set-Cookie: SID=31d4d96e407aad42; Path=/; Secure; HttpOnly
Set-Cookie: lang=en-US; Path=/; Domain=site.example

== User Agent -> Server ==

Cookie: SID=31d4d96e407aad42; lang=en-US

```

Figure 3 Set-Cookie Attributes

Multiple cookies can be stored with user agent. In Fig.3, Server stores SID with Secure and HttpOnly attributes ensuring added security for some cookies. Cookie lifespan is set by the Expires attribute. A common technique used by servers to remove a cookie is by providing it an Expires value with a date and time in the past. New cookies provided by the server will supersede old cookies. Thus, removing the cookie.

It is observed 90% of Tracking cookies have a lifespan of more than 24 hours and value longer than 35 characters (Bujlow et al., 2017). The long cookie value is due to the unique user identification value. To evade tracking detection, companies store the user identification in fragments within multiple cookies (Li et al., 2015). Also, a study from 2011 has shown only 30% of users clear their browser cookies within a month (Weinberger, 2011). Therefore, persistent cookies have proven to be one of the most efficient ways to track users.

The allowed cookie size limit per domain varies among browsers, 4KB is the usual limit for most browsers. However, some major browsers like Firefox and Chrome place no restrictions on the maximum size per domain.

The domain isolation property of cookies restricts servers from accessing cookies other than their own. Third-party companies can undermine this property by placing code in the website allowing themselves to track users across other domains. For example, Facebook like buttons can be seen across sites other than Facebook. This allows Facebook to track users across domains other than its own. It is common practice for websites to use analytics tools to track their site performance. The most widely used Analytics tool is Google Analytics, which is used by 85.6% of 56.1% of websites on the internet (W3techs, 2021). This allows Google to track their users on multiple platforms.

3.1.1.1.2 Flash Cookies

Flash cookies are essentially the storage mechanism known as 'Local Shared Objects' (LSO). It is meant for Flash Player browser plugin to store data on it. Although it was designed to act as audio/video cache, LSO can be used to store other data. Tracking companies took advantage of this feature by storing user identifiers. The main reason companies took this approach was because most browsers allowed users to delete HTTP cookies. Deleting flash cookies was often not seen as necessary thus adding one extra step to improve privacy and increasing chances of tracking users (Mohamed, 2009). In 2009, more than half of the Internet's top websites utilized Flash storage for web tracking (Mohamed, 2009). LSOs can store up to 100KB data which is 25

times more than what a normal HTTP cookie can store. This not only extends the limits of collection and storage of more data but, Flash cookies are shared by all browsers installed in the Operating System. This would mean even if user attempted switching profiles between various browsers, device could possibly still be identified due to the common user identifier present in the LSO.

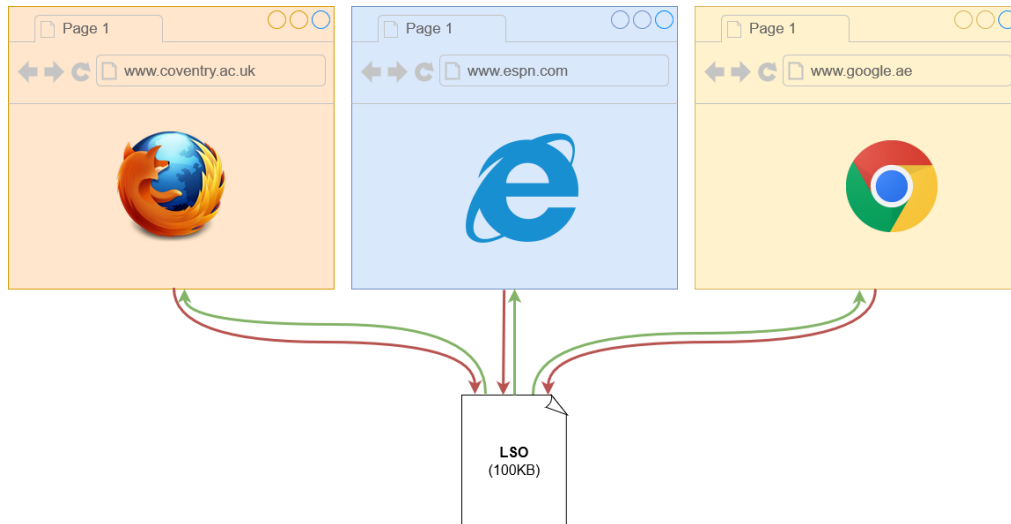


Figure 4 Local Shared Object Access

Fig.4 illustrates how multiple browsers access the same LSO stored on the user's device storage. Therefore, a site storing data on LSO through Firefox can also retrieve the same information if user visits the site through Chrome or Internet Explorer. Unlike HTTP cookies where storage is isolated. Microsoft Silverlight storage and Java Web App persistent storage use similar storage mechanisms which could also be used like LSOs. Threat posed by Flash cookies have declined in recent times because newer browsers allow deletion of Flash cookies. Also, Flash is technology that is being phased out. But most browsers still support Flash and could still be at risk to tracking using LSO if misconfigured.

HTTP Cookie respawn: due to the larger storage capacity LSO are able to store more complex data types. Hence, LSOs have been found to respawn deleted HTTP cookies to resume tracking (McDonald & Cranor, 2011). McDonald et. al also found just two instances of cookie respawn in the top 100 websites of 2011. It was found that flash cookies were utilized to revoke over 175 deleted HTTP cookies in 107 most visited sites (Acar et al., 2014). The use of Etags, HTML5 storage or Flash to revoke a deleted cookie results in the respawn of the cookie commonly known as evercookie. Websites that have deployed evercookies have a simple function. The unique ID generated for the user is stored in multiple locations such as cookie and LSO. In the earlier days, browsers did not offer the functionality to clear LSO or Flash cookies. Therefore, even if a user clears HTTP cookies, the stored value in LSO would be easily used to respawn the deleted cookie thus maintaining persistence. In addition to LSO, Indexed DB APIs which are storage mechanisms for NoSQL database for JSON have also been found to contain cookie respawning code (Acar et al., 2014).

3.1.1.2 Cache-based techniques

Cache-based web tracking is done by taking advantage of browser cache storage. When user visits a website, along with HTML other types of data are downloaded. This could be additional JavaScript files, images, etc. browsers have built-in storage mechanisms to prevent redundant downloads. This improves time pages take to load on second visit. Websites can also determine

if user has already visited the site before by considering where image is downloaded from. It could be assumed user has visited site before if image is pulled from cache or assumed first visit if downloaded from the server (Bujlow et al., 2017). This same principle is used by advertisers who have multiple objects on multiple websites and compare them to user's cached copies to reveal sites visited by user. The following paragraphs discuss how cache-based techniques have been used to track users.

The HTTP entity tag, ETag and the Last-Modified HTTP header are used to provide web-cache validation (Derksen, 2016). Consider, tracking object placed in website that would be downloaded to browser cache. The first download will have the following HTTP headers: Expires, Cache-control, Last-modified, and ETag.

Two key properties of Etag and Last-Modified that facilitate tracking:

- Etag field can store values up to 81,864 bytes, sufficient for storing long strings.
- Although Last-Modified header should ideally store only date and time values, it has been shown to accept any string value.

These properties allow both Etag and Last-Modified to contain user identifiers as values.

Assuming a user has already visited a site before, and the browser cache has previously downloaded object X from server. Upon second visit, browser sends HTTP request with If-Modified-Since and If-None-Match headers. Both headers contain ETag and Last-Modified values provided on last visit, respectively. Server can then compare the values retrieved from the cached copies to check if it still valid. If the cached copy is outdated, server sends a new document otherwise server simply sends a Not Modified status.

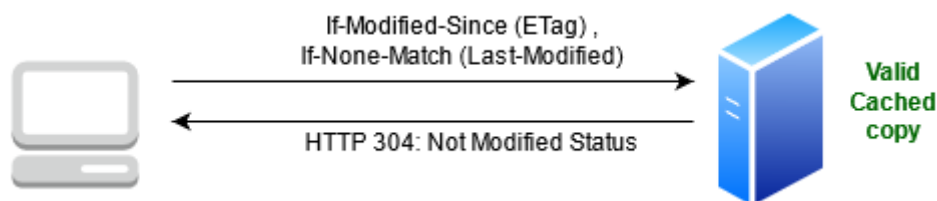


Figure 5 Valid Cache-copy

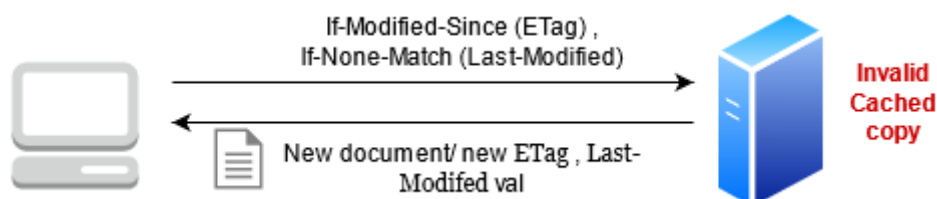


Figure 5 Invalid Cache-copy

Similar cached tracking can also be performed by using embedded identifiers. When client requests an HTML file, user identifiers stored in invisible div tags are sent. This is later referenced and can be read from browser cache by multiple websites.

Another way of using cache to track users is with the use of JavaScript. This would require browsers to allow execution of JavaScript to work. It can be used to calculate load times of images or any file that needs to be downloaded. By calculating and comparing load times, one can easily figure out if user has visited the site before as there will be significant difference between load times.

Consider the following example where *www.google.ae* was loaded for the first time in the browser. The recorded load time for image '*googlelogo_color_272x92dp.png*' was 263 ms. As shown in Fig.7. Visiting the page again however reduced the load time significantly for the same image due to the browser accessing that material from cache.

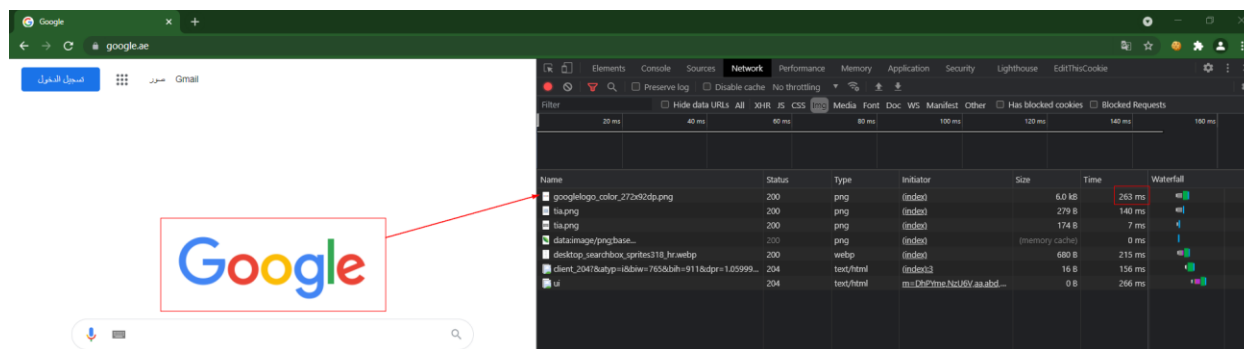


Figure 6 First visit image load time

Reloading the page causes browser to access cache storage where it has saved image temporarily. As seen in Fig.7 the response time has dropped to 0 ms. This drastic change in load times imply user is not visiting site for the first time.

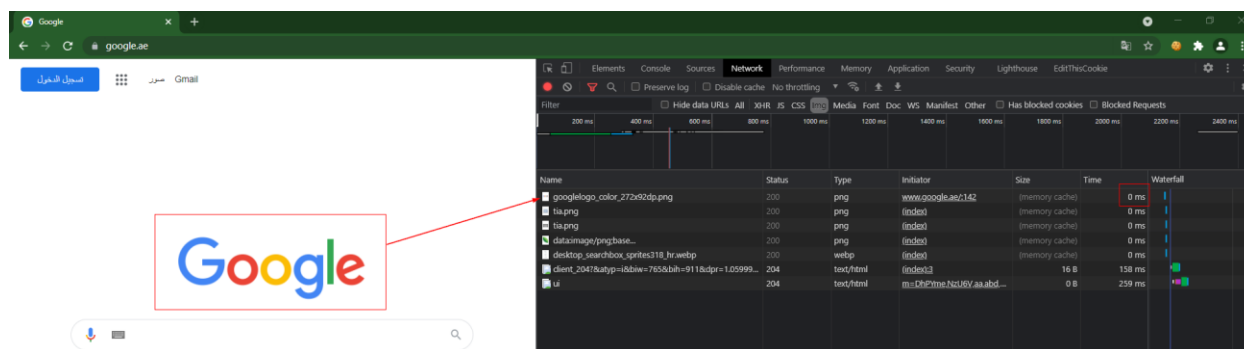


Figure 7 Image from cache load time

JavaScript can also be used to trigger a DNS lookup. The browser maintains its DNS Cache and based on its response time it can be inferred if browser is pulling that data from cache or if it is a website browser has not visited before.

3.1.1.3 Fingerprinting techniques

Apart from HTTP cookies, fingerprinting techniques are one of the most used stateless tracking mechanisms. Without the use of cookies, scripts run by the website can collect user data. When performed using multiple fingerprinting tactics, data sufficient to uniquely identify user can be collected. Fingerprinting is not only limited to browser details, there are several techniques that when used in combination could also help acquire data about device OS, time zone, installed software version, etc. This acquired 'fingerprint' passed through a fingerprinting algorithm serves as an identifier. The different kinds of fingerprinting discussed in lower paragraphs are: Device, Browser and Operating System fingerprinting.

3.1.1.3.1 Device

3.1.1.3.1.1 Network fingerprinting

Simple PHP script, see Appendix X. allows obtaining client IP address via incoming HTTP requests. JavaScript also can instantly provide the geolocation coordinates and local IP address with this information. In addition to JavaScript, Java applets can be used to execute browser functions. Java applets increase fingerprinting functionality by providing knowledge about device firewall status. As mentioned in the Java SDK documentation, applets embedded onto a page can invoke other applets on the same page. Although usually applet function is halted when user leaves page, applets do not have to stop running. In the Java SDK documentation,

there are a clear set of security restrictions that are imposed on Java applets. However, if these applets are loaded from client device locally, none of the restrictions apply.

The use of proxies to conceal IP address can also be detected with fairly easy methods by reading contents from the Forwarded HTTP request field. This HTTP header alone reveals the use of Proxy tracing back to original IP address as seen in Fig. 8

```
Forwarded: for=192.0.2.60;proto=http;  
by=203.0.113.43 Forwarded: for=192.0.2.43,  
for=198.51.100.17
```

Figure 8 Forwarded HTTP header

An extra step to detect proxy usage could be done using Flash. This is mainly because proxy configurations on user-end are merely just user preferences. Flash applications can choose to ignore those preferences. Flash and WebRTC applications do not prioritize proxy preferences as much as a typical browser dealing with mainly just HTTP. WebRTC applications facilitate communication between browsers and mobile applications, to enable this real-time communication WebRTC applications are allowed to request information from browsers. All the major browsers shown in Appendix X support WebRTC framework. In 2015, Dailymotion reported the seriousness of 'Geo-inference attacks' (Howell O'Neill, 2015). Geo-inference attacks use the same concept discussed in section 3.1.1.2 to obtain user's location. Almost 62% of Alexa Top 100 websites have been involved in such geolocation data leakage tactics (Jia et al., 2015). According to Yaoqi Jia et. al, based on load times tracking companies can fairly accurately track down user to their country, city and even neighbourhood.

3.1.1.3.1.2 Battery Status API

HTML5 provides a battery status API that allows developer access to device battery information. The main intention of this feature was to enable web applications to suitably adapt resource usage if noticed a shortage of power. The W3C had introduced specification stating this feature had minimal impact on privacy. Device can be uniquely identified over a short duration even while visiting multiple sites. To explain this further, consider a device whose battery information are collected from multiple different domains. The battery status API allows access to two vital pieces of information: chargingTime and dischargingTime. These two values are updated at the same intervals. Therefore, after every 5 minutes tracking scripts on different sites will read the same values. Tracking becomes easier when the same third party script is used in different domains. Moreover, battery status API does not require special permissions to access battery information. Browsers do not have to notify user about the application requesting to access allowing this data. This type of tracking, therefore, can be done in a stealthy manner. The level of accuracy is browser dependent. While majority of browsers display this value as just two digits, Firefox browsers can display battery levels as low as double precision.

3.1.1.3.1.3 Audio fingerprinting

The AudioContext JavaScript API helps produce an audio-processing graph. Similar to canvas fingerprinting, see Section 3.1.1.3.2.1, Audio graphs generated by different machines and browsers have varying differences that can be used to distinguish browsers from each other. Appendix X has a sample of the data collected to generate audio fingerprint. It is important to note that for audio fingerprinting, there is no collection of sound created or sound recorded from the machine. It is simply a fingerprint of the device audio stack. The Web Audio API does this by creating a 'context' within which the creation of nodes and the execution of audio takes place. Context contains three main components: source, processing, and output. The key property exploited for fingerprinting is OfflineAudioContext.

AudioContext needs to show the destination property of all the audio within the context.

The `OfflineAudioContext` functions slightly different from `AudioContext` with regards to how it generates audio. It does not create audio to the device hardware directly. Instead, it creates it and saves it to an audio buffer. Forcing the destination to be in-memory.

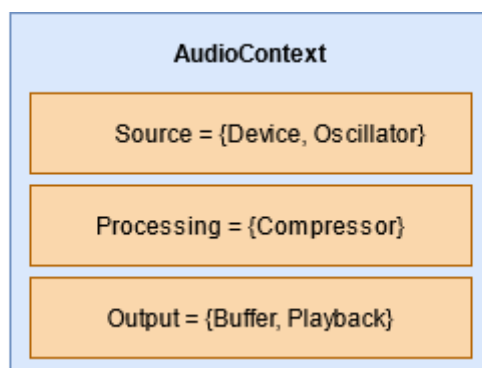
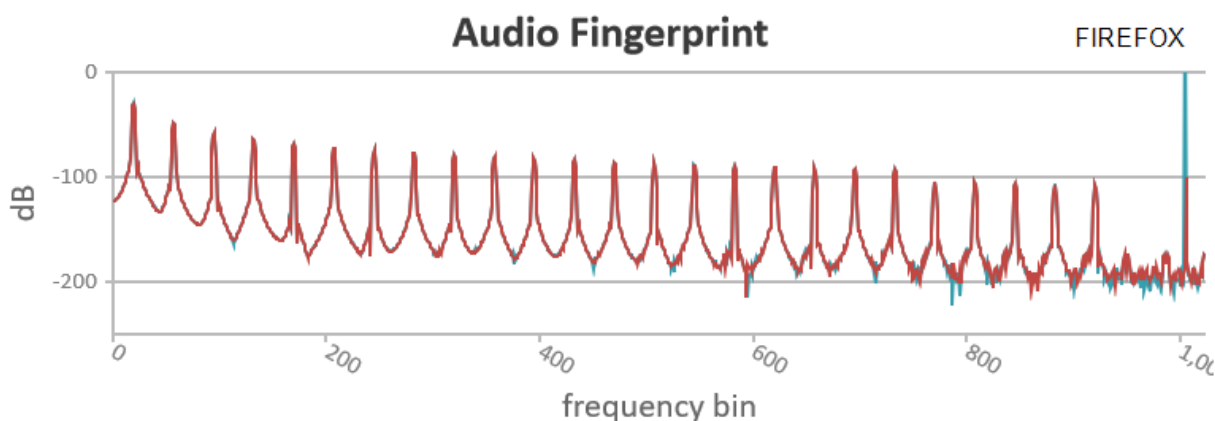


Figure 9 composition

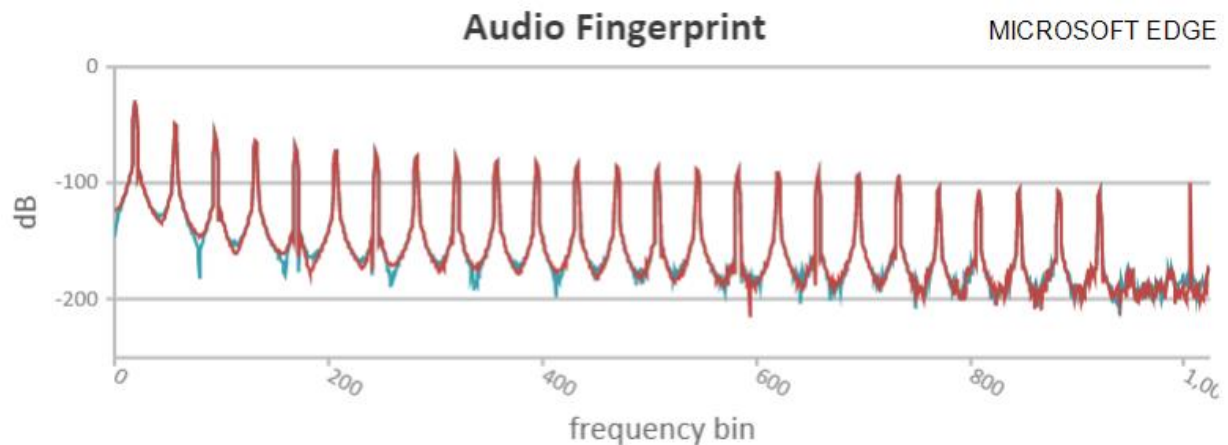
This `AudioBuffer` contains an audio snippet. Snippets in the buffer are generated by a source. Oscillators are good sources to generate audio as they employ mathematical functions to generate periodic wave functions. Processing of the snippet usually involves using function `DynamicsCompressorNode`. This function is responsible for the variation in snippets between browsers. Since hardware capabilities are different in different devices, it creates a slightly different output that can be uniquely identified.

A comparison of fingerprints obtained from Google Chrome, Mozilla Firefox, and Microsoft Edge all running on the same device is performed. Due to some browsers being unavailable on Linux, this test is performed on the host Windows machine. See specifications in Appendix X. Privacy researchers Steven Englehardt and Aravind Narayan have developed an online tool attempting to show fingerprint obtained using the `AudioContext` API. Using this tool in our test environment we obtain the following results:

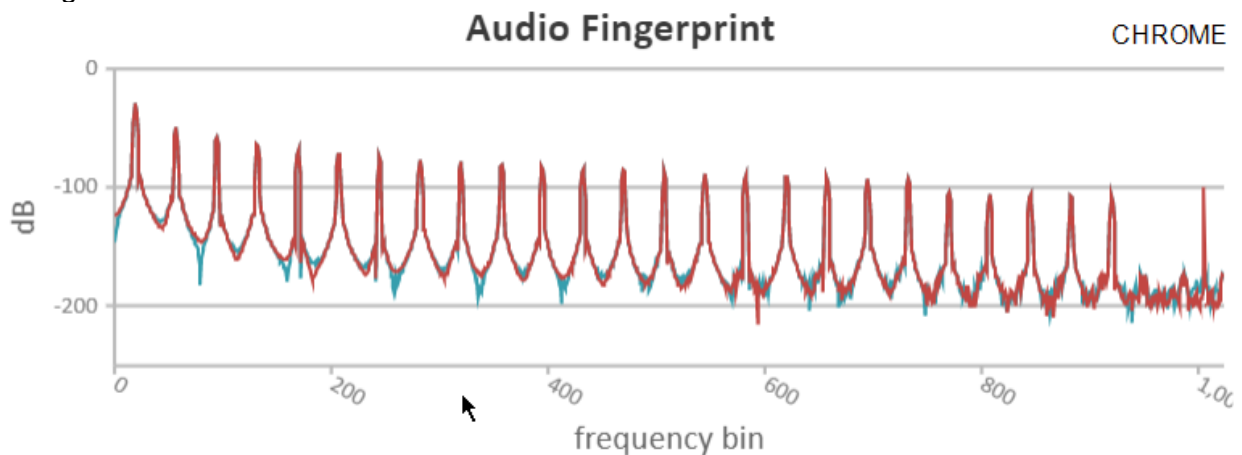
Firefox:



Microsoft Edge:



Google Chrome:



The fingerprints generated in these browsers are visually shown above. As observed, there exists slight differences between these fingerprints which add to the browser's uniqueness. These are fingerprints obtained purely through the AudioContext API.

Obtained fingerprints using only the DynamicsCompressorNode value from different browsers are shown in Table 2.

Browser	DynamicsCompressor	DynamicsCompressor Hash
Google Chrome	124.04347527516074	19f2ec826da994356fe069ffbebc1d80db815a8f
Mozilla Firefox	35.7383295930922	2dc43feaa1474319db71be0f4a9810c4a2a54524
Microsoft Edge	124.04347527516074	19f2ec826da994356fe069ffbebc1d80db815a8f

Table 2 AudioContext Fingerprints

As seen in Table 2, Google Chrome and Microsoft Edge share the same fingerprint value. This implies that although there is some degree of variation between browsers, when done on the same device deviation is not too great. Therefore, from a tracker perspective the combination of other techniques is necessary. This type of fingerprinting is possible only on browsers that support the AudioContext API. See Appendix X for list of supported browsers.

3.1.1.3.2 Browser

In 2010, Peter Eckersley conducted an experiment taking fingerprints obtained from approximately 500,000 browsers, this included time zone information, HTTP headers, installed fonts etc. and produced hashes of these fingerprints. The study showed 94.2% of fingerprints to be unique (Eckersley, 2010). This stateless tracking poses a greater risk to privacy as it disregards any safety measures that the user practices at their end since it does not rely on storing any data on the user device.

Peter Eckersley, in his study, used Shannon Entropy to better explain how fingerprint uniqueness is affected. By definition, Entropy of a variable is the average level of uncertainty possible in the variable's outcomes. Eckersley used the following formula to calculate entropy of a browser's fingerprint:

$$H(X) = - \sum_{i=1}^n P(x_i) \log_2 P(x_i)$$

Here,

A set of browser features $\{x_1, \dots, x_n\}$ where x is feature value and n is total no. of features
 $P(x)$ probability mass function

With the use of this formula, Entropy values can be calculated for a set of features of browsers. Browser features include but are not limited to date format, User-agent, Canvas fingerprint, DNT headers, installed fonts etc. not all features are equal and some features introduce more uniqueness to the overall browser fingerprint. According to Eckersley, the top attributes that generated the most entropy were from Javascripts. The higher the entropy value, the more unique the fingerprint. In 2019, Peter Hraška conducted an experiment to show the increase in entropy as more features were taken into consideration. He started out the experiment by taking a total of three features into consideration, namely, Date format, User-Agent and Available size. The initial entropy result was 14.2218. They continued increasing the feature subset size by including more features. A subset of 9 features resulted in entropy value 16.5168 .

Companies are aware of the capability of many browsers to manipulate their User-Agent field and claim to be a browser they are not. It is for this reason, User-Agent information is deemed unreliable. There are other methods used to obtain more accurate information.

HTML5 and CSS fingerprinting is done to obtain browser version. In CSS fingerprinting, the CSS filters, CSS properties and CSS selectors are compared and used to distinguish different browser versions. A simple JavaScript function, see Appendix X can reveal if a certain CSS function is supported or not.

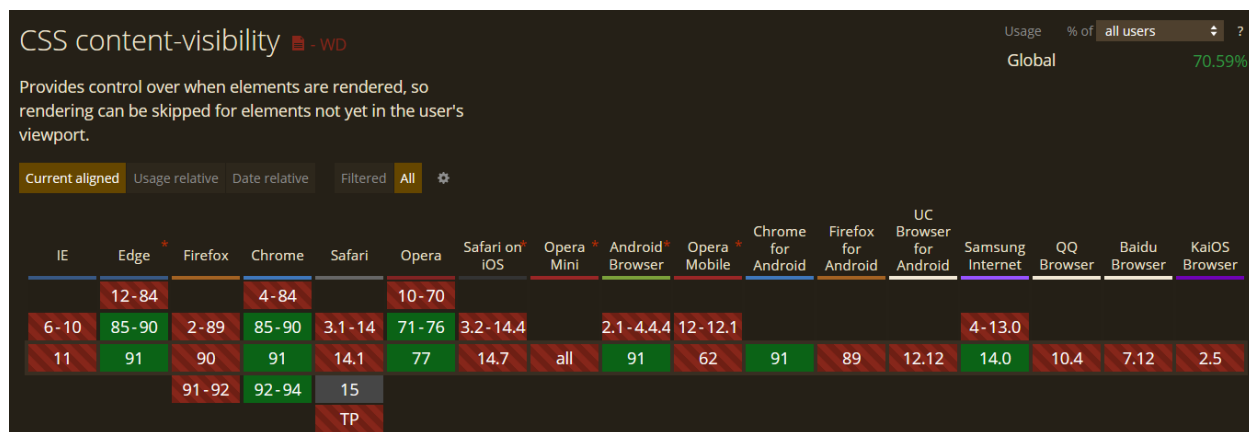


Figure 10 content-visibility browser support

Fig. 10 shows the browser versions that support relatively new CSS feature, content-visibility, based on this a fair estimate of browser version can be obtained. HTML5 fingerprinting takes a slightly different approach. With the recommendation of HTML5 in 2014 by W3C, came new APIs that improved developer features. In HTML5 fingerprinting, comparison between how the standard is implemented in browsers are done. Newer browsers will support newer tags. A total set of 242 tags, attributes, and features of HTML5 could be used to identify browser version (Unger et al., 2013).

3.1.1.3.2.1 Canvas Fingerprinting

Canvas fingerprinting is a common browser fingerprinting technique. This technique exploits the ability of browser canvas API to generate images and fonts on the system. Since the content generation ability directly depends on the underlying hardware of the device, there tend to be differences in the way those images are generated (Mowery & Shacham, 2012). For example, quality of the same 3D rendered image could be different if produced by devices with varying GPU capacity. This technique is commonly implemented using the HTML5 canvas element. The browser is requested to generate a graphic which is often made invisible hiding it from the user. Due to the differences in hardware, the generated graphic will slightly differ between devices, this uniqueness can be used as an identifier. See Appendix X for the JavaScript code using FingerprintJS library to generate graphic. Below mentioned are the list of factors graphic render depends on:

- Operating System
- Font library, installed fonts
- GPU
- GPU driver software
- Browser

Canvas fingerprinting on its own does not yield high entropy browser characteristics sufficient to identify user consistently and uniquely. Which is why it is typically combined with other forms of fingerprinting. Mowery and Shacham found that a browser will share its canvas fingerprint roughly 1 in 1,000 browsers.

3.1.1.3.3 Operating System

OS details can help collect more data points to increase tracking accuracy. Table 2 shows the various details that could be obtained using JavaScript and Flash plug-ins.

	OS Version	OS Architecture	System Language	User-specific Language	Local timezone	Local date and time	Installed Fonts	Color depth	Screen Dimensions	Audio capabilities	Webcam availability	Microphone access	Printing support	Read access device storage
JavaScript	✓	✓	✓	✓	✓	✓	✓	✓	✓	○	○	○	○	○
Flash	✓	✓	○	○	○	○	✓	○	○	✓	✓	✓	✓	✓

Table 3 OS Fingerprinting

In addition to these, ActiveX controls can also be used to obtain OS details. One study found BlueCava, a famous data broker to use ActiveX controls to retrieve desktop name, OS installation date and TCP/IP parameters (Nikiforakis et al., 2013).

3.1.1.4 Future of web tracking

3.1.1.4.1 Federated Learning of Cohorts (FLoC)

FLoC is Google's attempt to improve privacy regarding targeted advertising. At the time of writing this report, FLoC was Google proprietary technology and still in its early stages of testing. However, Google plans on replacing all third-party tracking cookies in the chrome browser by the end of 2023 (Bohn, 2021). Therefore, if FLoC improves privacy as it promises, there is a possibility to see mass adoption by other companies. As with any emerging technology, since the announcement of FLoC, it has received massive criticism from other pioneers in the field of privacy enhancement. A small percentage of Chrome users were given access to the initial testing of FLoC with restricted usage limits. Due to the lack of unbiased, real-world testing and performance evaluation of FLoC, the following paragraphs will take a deep look at the functioning of FLoC and review criticism received to uncover possible privacy-breaching flaws with the model.

3.1.1.4.1.1 FLoC: Overview

Federated learning of Cohorts browser API is a type of web tracking that aims to show relevant advertisements to users that are more likely to be interested in something without having to individually track them. It does this by learning from user browsing history and grouping them into clusters known as 'cohorts'. The idea behind cohorts is to minimize individual tracking with the use of cookies. Therefore, the main component of FLoC are the cohorts. More importantly, the quality of cohorts. This section will examine the process of building cohorts.

3.1.1.4.1.2 FLoC: Cohorts

To get assigned cohorts there needs to be a cohort ID generated for users based on their browsing history. The cohort ID generated will be used to assign users to a particular cohort. This ID assignment will need to take a few things into consideration.

- Users with the same cohort ID need to be grouped into a cohort where they share similar interests.
- To preserve privacy, assignment algorithms should not be supervised.
- Determining cohort to be assigned should be efficient and not very resource demanding.

It is important to understand the effect cohort quality has on privacy. For ensuring privacy, the cohort constructed needs to have a certain minimum number of users. Determining this number could present two challenges.

1. Large Cohort size: In this scenario, cohort has an excessive user population. Larger the user base, the more diverse the interests shared will be. Essentially making it difficult to group them into a cohort that shares a common interest. However, a large user base decreases the chances of being individually recognized which is good for privacy.
2. Small Cohort size: In this scenario, cohort has a relatively small user population. Smaller the user base, the more specific their shared interests could be. Thus, resulting in more accurate, personalised advertisements. However, a small size of cohorts increases the chance of individually identifying users which is not ideal for privacy.

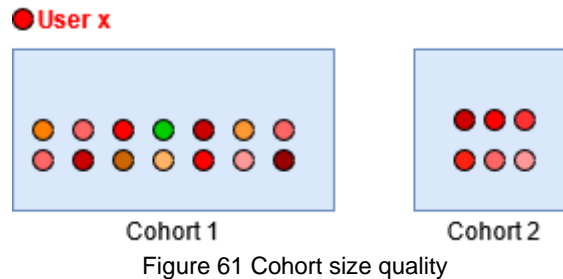


Fig.11 illustrates the difference between a large cohort size, Cohort 1, where diversity could challenge advertisement relevance versus a smaller cohort size, Cohort 2, where similar browsers are grouped together to show highly specific advertisements but can also increase the chances of user identification. Therefore, at some point, there will be a trade-off between privacy and utility.

An ideal cohort should comprise of large number of users with similar interests. As per Google's privacy sandbox requirements, cohort IDs are k-anonymous. Where, k is the minimum number of users cohort is shared with.

Therefore,

$$\text{Anonymity}(\text{Cohort id}) \propto k$$

where 'k' is minimum no. for cohort construction.

Greater the value of k, better the anonymity of the generated cohort ID.

Consider the following simplified example of cohort assignment to better understand it, Let us consider 8 distinct users with 8 different search interests.

Users = {User 1, User 2, User 3, , User 8 }

The sites 'www.pets.ae' and 'www.pet-help.com' cater to all the user needs and are the only sites visited by all of them. Cohort assignment will be affected by their interests and browsing history. We will consider both possibilities and review the output. Here, all 8 users will be divided into two cohorts. Users, their interests, and site visited are shown below.

User	Interest	Website visited
1	Dogs	www.pets.ae
2	Dog food	www.pets.ae
3	Cats	www.pets.ae
4	Cat food	www.pets.ae
5	Dog accessories	www.pets-help.com
6	Cat medicine	www.pets-help.com
7	Dog veterinary	www.pets-help.com
8	Cat accessories	www.pets-help.com

Table 4 Cohort Assignment

Assuming, cohort creation respects k-anonymity for $k=4$, Fig. 12 shows the possible outputs for cohort creation.

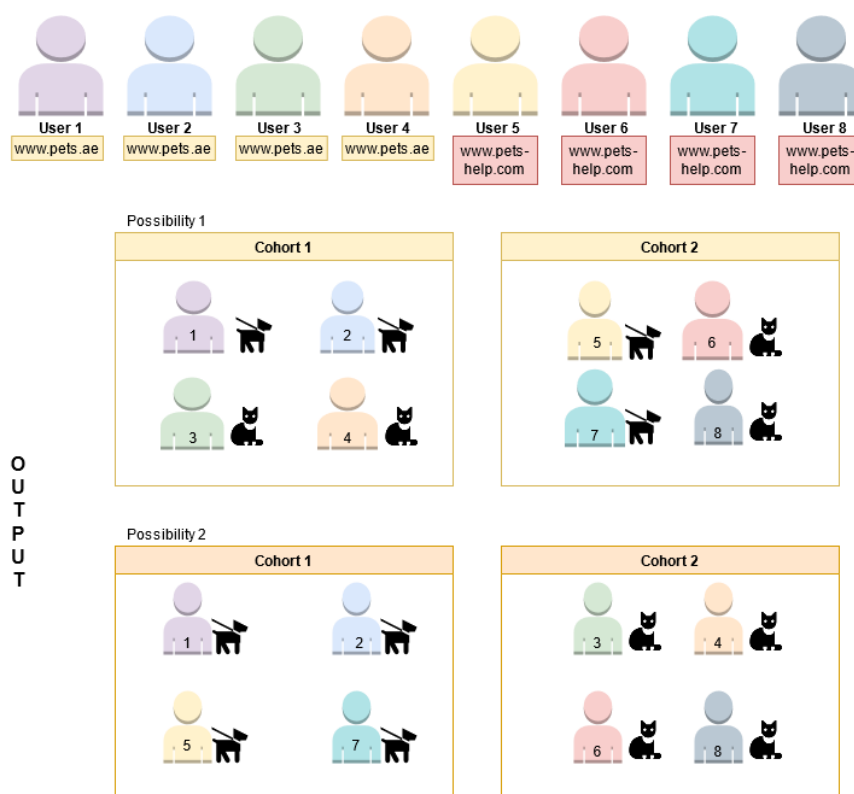


Figure 12 Cohort Assignment Possibilities

In the above example, the users are divided into two cohorts in two different ways.

Possibility 1, groups users based solely on browsing history, this results in a cohort that contains users with varying similarities.

Possibility 2, it takes a different approach and groups users based on interests. It groups users interested in dogs in Cohort 1 and users who like cats in Cohort 2. Grouping users minimizes probability of identification. But considering the above example, possibility 2 is more favourable for targeted advertising.

The second important function required for FLoC is the algorithm to generate the FLoC ID. For this FLoC uses PrefixLSH which is similar to the initially proposed SimHash algorithm. The browser is required to use every domain in the history to calculate a 50-dimensional floating-point vector whose coordinates are random. A server-side function would then be needed to calculate how frequently this 50bit hash occurs. Two big cohorts are initially formed one with hash value starting with 0 and the other with 1 and is repeatedly divided into cohorts containing at least 2000 users. What we understand from this working, as well as what is confirmed in the chromium privacy sandbox documentation is, despite being called 'Federated Learning', there is no federated learning and this is an unsupervised clustering mechanism.

Key aspects taken into consideration while cohort creation are k-anonymity and sensitive categories. Firstly, anonymity of users are required and ideally should not be uniquely identified based on data generated by FLoC. Secondly, users may not want to share some browsing activities especially some sites that reveal sensitive information. Google suggests tackling this issue by two approaches. Some sites that are known to be sensitive and belong to the sensitive category will not be included in the FLoC. Sites that are too similar to sensitive categories will also not be included in FLoC. Similarity is calculated based on usage statistics. For example, if 30% of users are known to visit sensitive category sites and 40% of users belonging to a certain site visit those sites, then FLoC considers that cohort to be interested in sensitive categories

and will not include it. FLoC only considers browsing history from the last 7 days as Google believes interests could change over time. Although FLoC promises a certain level of anonymity, when combined with other fingerprinting techniques, the anonymity offered greatly reduces.

4 Analysis of Current Tracking Technology

This section of the report aims to evaluate the current state of tracking by deeply analysing the discussed mechanisms in section 3.

4.1 Test Environment

The test environment was setup in an isolated environment using VMware. The virtual machine where all tests were conducted were Ubuntu.

Host operating system was a standard Windows 10 Home, 64-bit. For exact specifications of the virtual machines see Appendix F.

Tests, wherever applicable, were strictly limited to the following browsers:
Google Chrome, Firefox, Safari, Internet Explorer, and Edge.

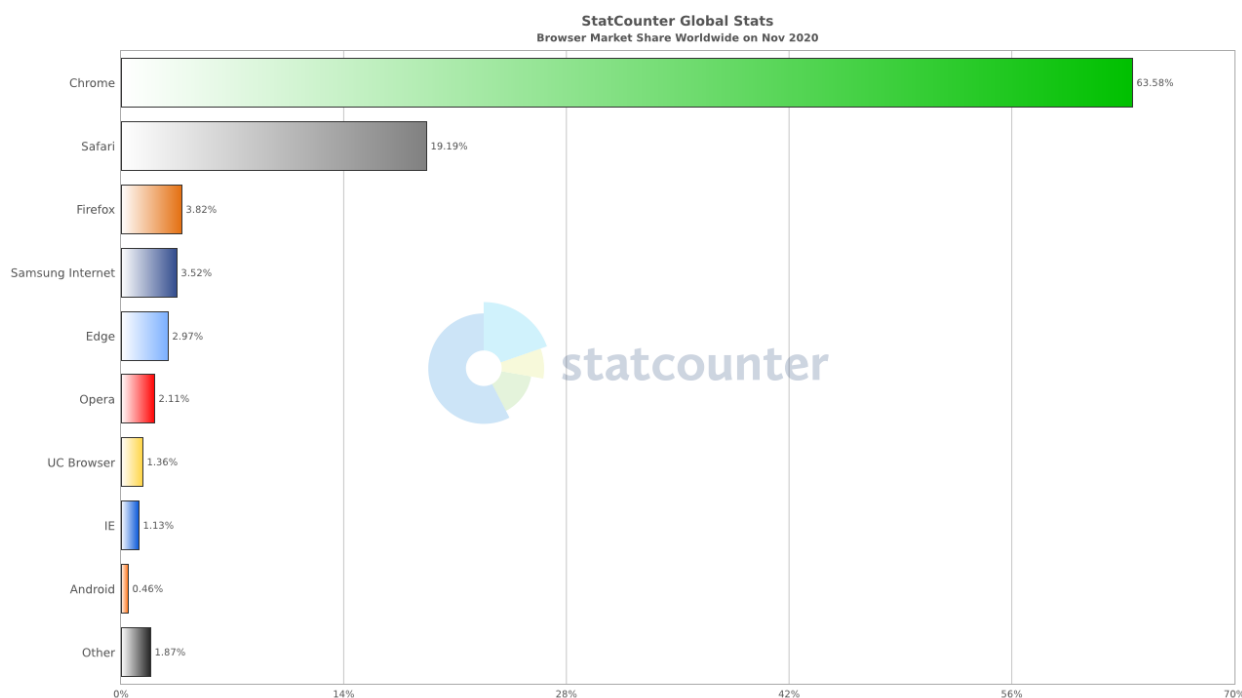


Figure 13 Browser Usage Statistics

These browsers were chosen based on the user statistic reports of StatCounter, W3Counter, and Net Applications of browser usage. See Appendix X for list of browser usage statistics.

A crawl list of 50 websites was created. The list was obtained from Alexa.com

The list obtained consists of the top 50 websites in the UK based on web traffic. See Appendix X for the list.

4.1.1 Auditing tools

Auditing tools that were used to measure performance of browsers and extensions:

Privacy report

Open-source browser extension that successfully detects third party cookies presenting the path, name, and value of cookies. From our test runs, privacy report has been consistent with the results. Privacy report also detects Fingerprinting scripts. Developed by Michael Cann, privacy report repository is available on Github. See Appendix X for the URL.

Orbis Eye

Chrome extension that generates a visual representation of all detected third party trackers.

4.2 Analysis: Cookies

This section aims to assess the current level of tracking done by well-known websites. We use the Alexa generated UK top 50 crawl list to visit 26 websites on the list. Websites featuring explicit content like pornography were avoided. Fig 15 shows the visual representation of encountered trackers by simply loading their webpage. This test was performed using bare Google Chrome with no privacy enhancements.

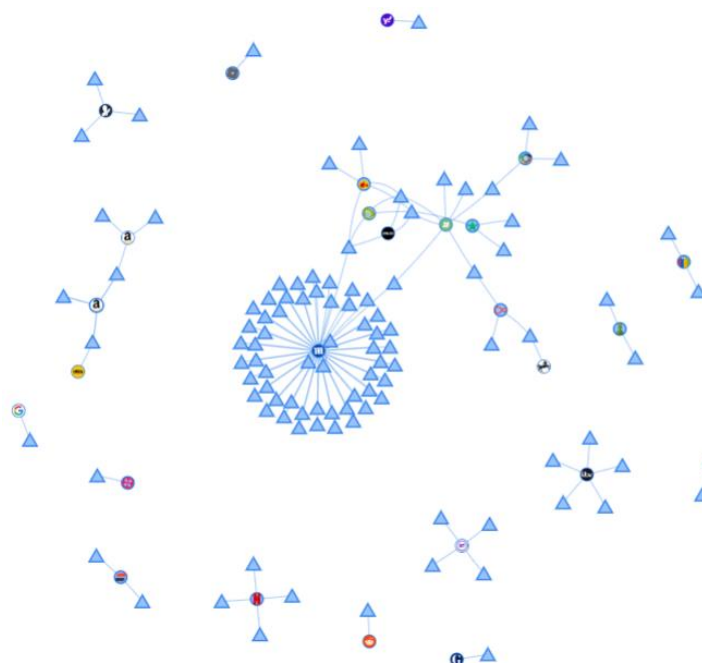


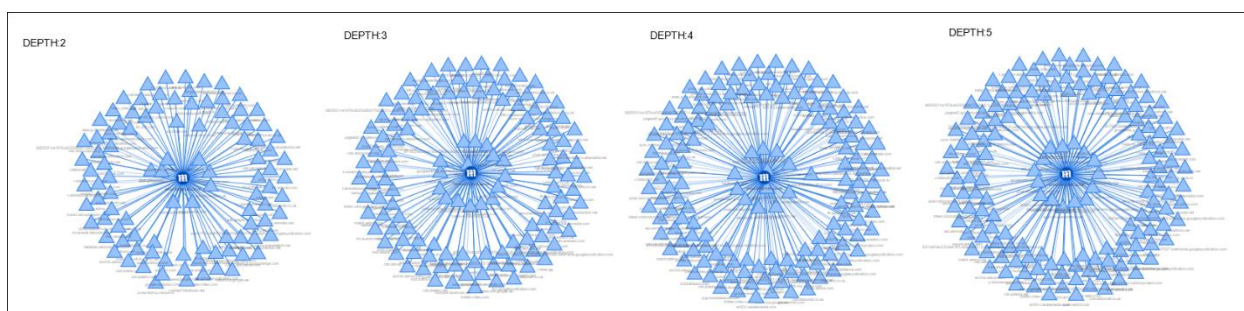
Figure 14 Third-party trackers in Top UK websites

Upon loading the 26 websites, it was observed that on an average a website is likely to have 4.65 number of connected third party websites, dailymail.co.uk being the highest having 56 number of connected third party websites. The minimum number of connected third party websites a website observed is 1. See Appendix X for detailed results. Observing the connected trackers, Google-related web trackers were frequently recurring. We see 12 trackers from Google. Google by far has the most widespread trackers on the internet with around 75% analytics trackers deployed in the top million websites (Weinberg, 2021).

Upon the initial cookie assessment that gave us a brief overview about the third party distribution in the UK's most accessed websites, we begin the second attempt to take a deeper look at the level of third party tracking.

In this test, we setup crawls with depth values 2,3,4, and 5. Crawl depth is the extent to which we crawl the website to discover third party tracking. Cookies and fingerprinting scripts are detected. We begin our analysis with dailymail.co.uk, the website with the highest number of trackers on our list.

We assume the typical user is going to do more than just load the website. Interacting with the website by navigating through web pages is simulated using different depth values. The average usage session in a website could vary significantly depending on the site. Content-heavy sites are assumed to have longer sessions since it is more engaging. On first visit, we observe a total of 114 cookies. Fig 16 shows visual comparison of third party trackers as we crawl deeper. Increasing depth values from left to right.



Crawls originate from the same root i.e., the main page of the website. This is typically the *index.html*. The results from our first one-domain crawl are shown below:

Depth	Cookies
2	165
3	201
4	204
5	210

As observed from Table 5, the majority of third party cookies are detected upon initial visit and further crawling only slightly increases cookies. We now conduct this experiment with a larger sample size, we use the same list obtained from Alexa subtracting the inappropriate websites.

We use a total of 45 different domains in the crawl list each of which set as a root to initiate our crawl. Due to the massive set of domains, we limit our depth to 2. The test was conducted over a period of three days. It is to be noted that some sites may simply present a login page at first visit expecting user to log in to use their services. These sites may not provide us enough information about their actual cookie usage since the crawl does not log in to a user account. A total of 868 third party cookies were detected from the crawl. Cookies had varying expiries ranging from session-only to 2 years. There were four notable cookies however from [ladbible.com](https://www.ladbible.com) that had an expiry of 7984 years. Inspecting the values of these cookies do not suggest they contain any unique identifier but it shows us that websites are able to set extremely long expiry durations.

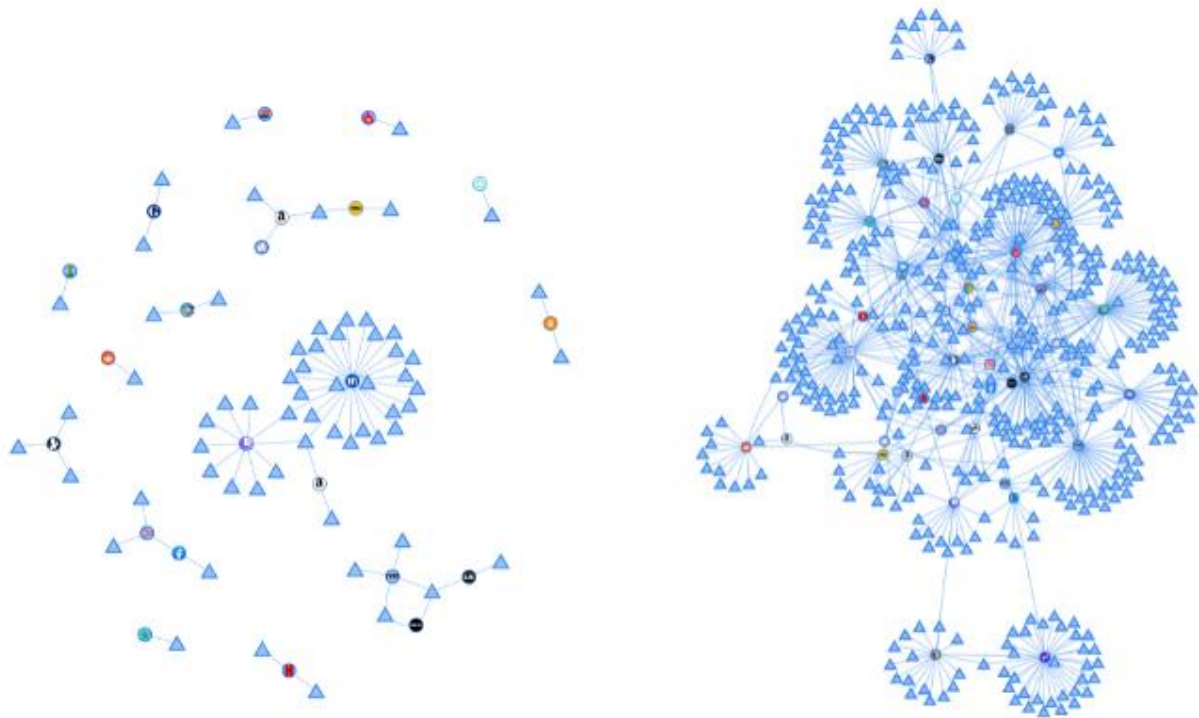


Figure 17 Before vs After Crawl

Fig. 17 shows us the cookie distribution after a depth 2 crawl. The full report is available on Github see Appendix X.

4.3 Analysis: Fingerprinting

For tracking purposes, it is important to understand that collecting just one fingerprint for one device will serve no good. Likewise, collecting too many fingerprints from just one source, say, one particular website will also not provide enough user information. Therefore, tracking is widespread. As mentioned earlier in Section 3.1.1.1, companies like Facebook and Google accomplish this by embedding proprietary code onto other domains to increase their fingerprinting area. The following sections attempt to analyse the current state of fingerprinting. Privacy report identifies fingerprinting scripts embedded in websites. Similar to our earlier experiment carried out in the previous section, a crawl is initiated through the UK Top 50 domains in attempt to detect fingerprinting. The use of APIs to recall identifiable information is identified. The complete set of APIs detected are available in the report see Appendix X. The initial crawl with depth value 2 identified a total of 116 fingerprinting attempts. The most recurring API detected was `window.navigator.appVersion` which returns the browser version.

4.4 Analysis: FLoC

FLoC analysis is based on theory. Since it was proposed as a replacement to third party cookies, we begin analysis by comparing it to cookies. If FLoC does not improve the overall privacy of users there should be no reason to switch to this technology.

At present, third party cookies do not have access to a user's browsing history. Cookies embedded onto a domain are limited to collecting user data from the same domain. But until now third party cookies have managed cross-domain tracking as discussed in previous sections. This shows us that there needs to be some dependence and cooperation between the trackers i.e., observers. Since FLoC, has access to user browsing history, it would ease one of third party cookie's main challenges. Two trackers may choose to cooperate and share collected data or decide to function independent. The current ad ecosystem already does this and is hugely dependent on it. In comparison to cookies, FLoC will give trackers more data. FLoC will first be introduced only to Google Chrome. Assuming that FLoC will be widely adopted at some point, there may also be a brief period where other browsers will accept both kinds of tracking. Therefore, if browsers do not have a strict cookie policy, combination of tracking with FLoC will result in a greater privacy breach.

One major challenge with the introduction of FLoC will be the threat posed by already existing fingerprinting techniques. FLoC proposal promises to offer an individual user belonging to a cohort of size k , k -anonymity. However, when accounting for the threat posed by fingerprinting, the anonymity offered significantly reduces. At present the minimum size of a cohort is 2000 users, 11 bits. Which means for a cohort of size 2000, 11 bits of fingerprinting entropy would suffice to identify users uniquely. Obviously, this will become increasingly difficult as the total number of users in the cohort increase. It is very unlikely that FLoC will exist without fingerprinting. In addition to browser history that is directly obtained, user identity could also be passed through sites completely based on user navigation. A website is able to gather information of previous website based on the URL and also through HTTP referrer. This is important to consider because FLoC IDs could also be transferred within sites.

A potential vulnerability is Sybil attack. Where one bad actor could counterfeit the identities of other users. The attacker upon identifying a cohort could increase the size of the cohort. This will result in extending the prefix for that particular cohort. As discussed earlier, the increase in prefix will cause collection of more specific data of the users. It is also known that cohort sizing will be done server side and this requires the browsing history to be unencrypted. Which presents a security issue.

Examining how FLoC may handle sensitive categories uncovers another problem. At present, Google categorises some content as sensitive which is used by Google Ads. However, one such problem with this is, sensitive is subjective. What one user finds sensitive, the other may not. To have one universal sensitive list is a challenge of its own. FLoC's attempt of trying to identify sensitive categories based on similarity discussed in section 3.1.1.4.1.2 presents an additional problem. If a user is in a sensitive cohort, the FLoC assignment for that would be an empty string. This does not reveal what sensitive category user belongs to but it gives us information that user may be interested in something sensitive. For example, health-related websites may be seen as sensitive. An insurance provider could therefore know a potential client is in a sensitive cohort. Such practices have been done before with third party cookies and could possibly be done again with FLoC.

Part - II

5 Preventive Measures: Testing

Part I of the report shows the various tracking methodologies adopted by companies so far. With a deeper understanding of the techniques and methodology of web tracking, each of the above discussed can be tackled with some measures. The analysis of the techniques performed in section 4 provide us with a benchmark to compare our proposed tools with. All comparisons shall be conducted in hopes of improving performance and privacy. We start off this section by discussing tools needed to address each tracking technique in the same order they were first discussed.

A total of 5 browser extensions were chosen to be analysed and compared. In the following paragraphs, the extension functions are studied and tested on compatible browsers. The testing of these extensions were only limited to the browsers mentioned in Section 4.

5.1 Browser

As seen in Fig. 13, majority of desktop users use one of four browsers: Chrome, Firefox, Safari, and Edge. Browser selection is dependent on the availability of add-ons and extensions. As these extensions will allow for better privacy. While evaluating privacy protection, it is also crucial to consider performance as that hugely affects user experience.

Table 5 shows the browser engines of the chosen browsers. Trident and EdgeHTML are the two Microsoft proprietary browser engines for browsers Internet Explorer and Edge, respectively.

Browser	Browser Engine	License
Google Chrome	Blink	GNU LGPL
Mozilla Firefox	Gecko	Mozilla Public
Internet Explorer	Trident	Proprietary
Microsoft Edge	EdgeHTML	Proprietary
Brave	Blink	GNU LGPL
TOR	Gecko	Mozilla Public
Safari	Webkit	WebKit

Table 6 Browser Engines

All browsers in Table 5 have a private mode. This mode is called 'Incognito' in Google Chrome, 'InPrivate' in Edge and Internet Explorer, 'Private Browsing' in Safari and Firefox, 'Private Window' in Brave. In comparison to the standard mode, it stores less data locally and is usually deleted as the user closes the browser. When in private mode, most browsers treat it as a fresh session and allow no access to older data such as history and cookies. Fingerprinting scripts do not discriminate between these modes and simply using private mode will not significantly improve privacy. But, due to the lack of access to previously stored user data such as HTML5 storage, fingerprinting becomes less accurate.

We begin evaluating browser based on their ability to access data created or stored in different modes. Privacy is better when the data generated in one mode is more isolated.

All browsers were populated with some user data in standard mode first and was later attempted to be accessed from private mode. Table 7 contains summary of the experiment carried out.

	Chrome	Firefox	Edge	Safari	IE	TOR	Brave
Cookies	○	○	○	✓	○	○	○
Browsing history	✓	○	✓	✓	○	○	✓
LSO cache	○	○	○	✓	○	○	○
Bookmark	✓	✓	✓	✓	✓	✓	✓
Download list	✓	○	✓	✓	○	○	✓
Search history	✓	✓	✓	✓	✓	✓	✓
In-browser cache	○	○	○	○	○	○	○

Table 7 Private mode vs standard mode access

Most browsers disable extensions and add-ons in private mode, by default. They do this as a safety measure because some add-ons could leak browsing data either locally or to websites, subverting the effectiveness of private mode. Therefore, one should review the add-ons they use in private mode. To be used in private mode, add-ons will need to be manually granted exception. Gaurav Agarwal et. al developed Firefox extension '*ExtensionBlocker*' that disables all unsafe extensions with regards to Privacy (Agarwal et. al,2010) discussed in later sections.

The qualities of a good privacy-respecting browser are as follows:

- Dedicated Private mode.
- Ability to clear cache, including Flash and HTML5 storage.
- Built based on a well-recognised, preferably open-source browser engine.
- Availability of privacy enhancing extensions.
- Provide users option to send DNT requests.

It is not uncommon for privacy conscious users to have multiple browsers installed on their machine. Using different browsers for different tasks has its own advantages. This is further discussed in section 5.3.1 where the idea of compartmentalization is introduced.

5.2 Browser configuration

Upon choosing a browser that, at the bare minimum, allows privacy features discussed in the previous section, it is important to configure it accordingly. A majority of websites now depend on JavaScript for the dynamic operations of their site. Although JavaScript is increasingly used for fingerprinting purposes, simply disabling JavaScript will not solve the problem. Disabling JavaScript could cause some web applications to misbehave or crash. Newer browsers thus enable JavaScript by default. A better way to manage JavaScript execution perhaps could be done using a JavaScript manager like NoScript. The extensions chosen for examination are all available on their respective app marketplace which the average user can easily access and install.

The recommend two types of extensions to be considered for better privacy can be classified as:

1. Execution Blockers
2. Third-party tracker Blockers

While some browsers may already be capable of performing functions of these extensions, a dedicated extension is simply more updated and easier to manage.

DNT headers can be enabled by the user. It is a HTTP header that submits a request to the website asking not to be tracked. All browsers mentioned earlier support this feature. However, this hugely depends on the website and their desire to be respectful of user privacy preference.

SSL enforcers are also popular extensions that force HTTPS connections wherever possible. HTTPSEverywhere is one such well-known extension developed by the EFF. Newer versions of all major browsers provide this as an in-built feature and need no additional installation of an extension.

In a 2019 study, Johan Mazel et. al categorize privacy preserving browsing extensions based on their primary functioning mechanism into three main categories. Mazel et. al classify them as blocking lists, heuristics, and indiscriminate blocking (Mazel et. al, 2019).

Below are the list of extensions that were studied. The criteria used to select these extensions were: easy availability, availability in major browsers, and the user community size. The user community size was based on total number of downloads from the marketplace. Community size was taken into consideration as larger the user-base, the better support offered by the developers and the software is generally more frequently updated.

	Google Chrome	Brave	Mozilla Firefox	Microsoft Edge	Microsoft IE	TOR	Apple Safari	no. of users
Adblock Plus	✓	✓	✓	✓	○	✓	✓	10,000,000+
Ghostery	✓	✓	✓	✓	○	✓	✓	2,000,000+
NoScript	✓	✓	✓	✓	○	✓	○	100,000+
Privacy Badger	✓	✓	✓	✓	○	✓	○	1,000,000+
uBlock Origin	✓	✓	✓	✓	○	✓	✓	10,000,000+
WebOfTrust	✓	✓	✓	✓	○	✓	✓	1,000,000+

Table 8 Add-on availability

Table 6 shows the availability of all add-ons in different browsers. Number of users is calculated across all browsers based on data collected in 2021. Internet Explorer is an obsolete browser with the worst availability of add-ons.

5.2.1 Execution Blockers

As mentioned earlier, these are browser extensions that manage the execution of scripts on the website. NoScript is available on all major browsers. It provides the user better visibility and easier manageability of JavaScript, Flash, Java, and any other known plugins.

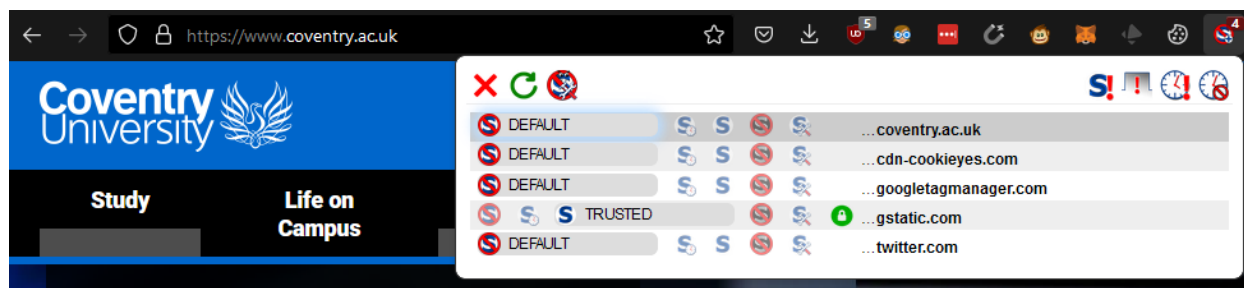


Figure 18 NoScript

NoScript indiscriminately blocks JavaScript that is not present in its 'Trusted' list. Obviously, this affects browsing experience as a lot of websites heavily rely on JavaScript. It can manually be set to Trust or Un-trust objects, media, frame, font, webgl, ping, noscript, unrestricted CSS etc. NoScript reduces on an average reduces the total number of third party HTTP requests to 87% (Mazel et. al, 2019). Krishnamurthy et. al use 'privacy footprint' to evaluate browser extensions. The interactions between first party and third party are plotted on a graph to get an idea of the level of privacy. Third parties are considered potential trackers. Mazel et. al performed a crawl of top 1000 Alexa websites using NoScript as an add-on with Firefox. They found that on an average NoScript reduces first party links to the top 10 third party trackers to below 600, see Appendix X. They also found that, out of 21 configurations which consisted of different combinations of extensions and browser settings, NoScript was out-performed by only RPC. Proving NoScript to be one of the best extensions for privacy.

We test the efficiency of NoScript by comparing it to the results obtained in Section 4.3. Conducting a crawl with NoScript enabled and default configurations brings down the total fingerprinting scripts from 116 to 8. This is a significant reduction of 93.1034%. Complete test results are uploaded on the Github repository see Appendix X.

5.2.2 Third-Party Tracker Blockers

This section examines Adblock Plus, Ghostery, uBlock Origin, Privacy badger, and WebOfTrust. The first three extensions have similar underlying working principle. They require a blocking list which uses regex-based rules on given domains. The lists are typically maintained by their respective developer teams.

Adblock plus

Is an open-source extension available on most major browsers. Adblock plus functions based on a list. Which provides users the ability to also whitelist some domains allowing some advertisements. This ability to hide certain HTML elements gives Adblock plus control over what images to show and what to hide. It is not limited to just images, Adblock plus can also hide iframes, flash embedded elements and scripts that have blacklisted sources. This also enables user to filter out advertisements and trackers based on custom rules. In addition to custom rules, users could also choose to use filtersets which are made available for users to subscribe. EasyList is one such well-known list that works with Adblock plus.

Ghostery

Is a Proprietary browser extension developed by David Cancel. The initial setup involves choosing from four different blocking types: 'Block default', 'Block nothing', 'Block Everything', and 'Choose from List'. By default, Ghostery blocks 1453 Advertising trackers, 530 Site analytics trackers and 20 Adult advertising trackers. These are all contained in Ghostery's community maintained list of over 2400 trackers. This initial configuration may not be something the typical user may want to do. From our test results, we see that Ghostery compares similar to

uBlock origin due to its enhanced list. Ghostery and other extensions mentioned in this section do not perform as good as NoScript but they also do not affect webpage quality much.

uBlock Origin

In comparison with the other two list-based third party blockers, uBlock origin is relatively newer. uBlock Origin by default contains the best lists offered by Adblock plus. Lists such as EasyList, EasyPrivacy, URLHaus, and other well-maintained lists are available with no additional steps. This makes the initial setup of the extension fairly easy. uBlock origin also has all the other features of Adblock mentioned earlier. One of the key features that uBlock origin promises is better utilisation of resources which result in better performance compared to its competitors. Therefore, if device hardware capabilities are a constraint, uBlock origin could be better suited than other competitors. From the tests performed later in the report, we see that uBlock origin outperforms both Ghostery and Adblock plus with default configurations.

Privacy badger

Privacy badger is a great tool developed by the Electronic Frontier Foundation. It is also open-source and available on most major browsers with the exception of Safari. In comparison with the three previously mentioned extensions, privacy badger uses a different working mechanism. Privacy badger attempts to study the working of a website to identify any tracking. This heuristic based approach requires some live data. Privacy badger is trained by constantly monitoring the cookies being accessed, if the same domain accesses a cookie as third party more than three times, it assumes tracking to be done. Privacy badger will then block that domain. In addition to that Privacy badger uses DNT HTTP headers to remove any revealing data in the referrers requests. Due to its working, privacy badger requires a period of training to study and block domains. Although with the right training Privacy badger could become very efficient, this may not be suitable when third party blocking is expected right away. The quality of training also greatly affects the efficiency of third party tracking.

WebOfTrust

WebOfTrust was a chosen candidate for the list of recommended extensions due to its popularity, availability across all browsers and remarkable achievements. It was a recommended choice by many industry experts in its early years. However, WebOfTrust has since then lost its user base. There have been other effective extensions that replaced WebOfTrust. It is important to mention the working of WebOfTrust because it uses a different working principle and it gives us an idea about why this working model may not perform well. WebOfTrust primarily works by rating websites. This rating is influenced by users and some blacklists. The sites are rated on their reputation and categorized into red, yellow, and green. Red being the least safe and green, safest. Since WebOfTrust relies on ratings provided by users, it has been observed that some sites have managed to attain a bad reputation despite not doing anything unacceptable. The forums of WebOfTrust community have shown users target a particular website to damage its reputation. This extension that once performed well, relied on its users too much. Which hugely affected its effectiveness. This shows us the importance of a good quality source that ultimately influences the blocking done. It is for this reason WebOfTrust and similarly functioning extensions will not be recommended.

Our results from section 4.2 were the baseline these extensions were compared against. Ghostery, Adblock plus, uBlock Origin, and Privacy badger were each individually tested under similar conditions. A total of 45 domains were crawled and the observations are shown in table below.

Extension	No. of third party cookies	Percentage change
uBlock Origin	241	-72.235%
Ghostery	305	-64.8618%
Privacy badger	347.6 (average)	-59.9539%
Adblock plus	411	-52.6498%

Table 9 Third party cookie blocking comparison

From our test, we conclude that with no additional steps to configure, uBlock origin is the best performing. It blocks 72.23% of third party cookies responsible for tracking. This could be due to its inclusion of comprehensive lists by default. Privacy badger was tested a total of three times because it requires a brief period of training. The total number of cookies detected however remained same after the third run. Adblock plus was the least efficient mainly due to its default list. By default, Adblock plus uses EasyList and nothing else.

We also examine the effect of extensions on fingerprinting scripts. From our test results in section 4.3, we find a total of 116 fingerprinting scripts across the domains. Table 10 shows the observations from our tests.

Extension	No. of Fingerprinting scripts	Percentage change
uBlock Origin	64	-44.8276%
Ghostery	66	-43.1034%
Adblock plus	86	-25.8621%
Privacy badger	104	-10.3448%

Table 10 Fingerprinting blocking comparison

Based on our test results, we conclude that uBlock origin performs well even in reducing fingerprinting. By reducing the overall detection of fingerprinting by 44.8%. Privacy badger initially returned 178 fingerprinting scripts which was more than the bare configuration of 116. This may be due to privacy badger checking the compliance of EFF DNT of third party domains. When configured not to send any GPC requests, we obtain a more reliable result.

5.3 Browsing behaviour

Privacy precautions do not end with just installation of tools. Successfully protecting privacy is greatly dependent on how one uses the internet.

5.3.1 Cookie Consent

Ever since GDPR was introduced in 2018, most websites catering to European users began showing cookie consent upon user's first visit to their site. This was initially a good initiative as it was supposed to give users better control over collection and processing of their data. However, this has fatigued users overtime causing them to accept cookie tracking without understanding the reason and usually against their desire. Websites too have been notorious in their consent notice design. As outlined in 2019 paper Christine Utz et. al show the common practices of websites to lure users to accept more than necessary cookies. Properties of a consent notice such as position, size, choices allowed, and the ability to use website before deciding on consent all influence the user's decision to accept cookies to some extent.

Position: There are usually seven different possibilities for the positioning of a consent notice. Top, bottom, middle, upper-left corner, upper-right corner, lower-left corner, and lower-right corner. It is found that bottom positioned consent notices are the most common, around 57.9% (Utz et al., 2019). An experiment conducted by Christine Utz et. al on a website that received 82,890 distinct users showed as follows:

- Desktop users, only 4.6% of user declined cookie consent, 7.8% accepted and 87.5% performed no action.
- 14.1% of desktop users interacted in some way with lower-left bottom positioned notices. This is the highest percentage of interaction received by one given position.

According to Utz et. al, a possible explanation for bottom positioned notices to receive greater interaction could be the obstruction of the webpage content. Hence, users are forced to interact with these notices to view the main content better.

Size: Typically, size is responsive to design and it adapts to area covered. As inferred from an earlier explanation, users tend to interact when the consent notice obstructs the page content. Therefore, bigger the notice size, the more likely it is to receive interaction from users.

Choices: Consent notices also provide users with list of trackers that user could allow or disallow. There are five main approaches to this. (i) No option, just mere information that trackers are being used and no options are given to the user to choose. It is assumed user agrees to all the site's tracking. (ii) Binary, only accept and deny choices are given. (iii) Confirmation-only, only some form of accept choice is given. (iv) Vendor, a more improved approach that lets users see and control which third party interactions will take place with. Giving user better control. (v) Categories, provides user with all categories of cookies being used. Categories commonly seen are functional, performance, and analytics. Category (ii),(iv),(v) provide users with the ability to deny tracker cookies. This is important to note because legally GDPR requires websites to recognise only active consent. Whereas many websites have shown to consider user ignoring cookie consent notice as acceptance from user (Nouwens et al., 2020). Nouwens et. al also observed many consent notices to pre-tick certain boxes increasing the likelihood of user accepting cookies.

Blocking: 7% of notices make the site completely unusable by either blurring, dimming or being non-responsive unless user makes a decision(Utz et al., 2019).

The studies conducted on cookie consent notices thus show that these are increasingly being used in a manner to influence users to accept tracking cookies. It is, therefore, important to manage cookie preferences suitably to protect privacy.

5.3.2 Search Engine Preference

According to statcounter, as of June 2021 87.76% desktop users use Google as their preferred search engine. Google is by far the most used search engine globally. Along with Google other search engines that are used are Bing 5.56%, Yahoo 2.71%, Yandex 0.8%, Baidu 0.63%, and DuckDuckGo 0.82%. Search engine preference is crucial to protecting privacy as most search engines are usually the starting point to a user's browsing session. The main purpose of a search engine should be to provide good-quality search results, unbiased and relevant. A search engine is where many users share their intimate information. Therefore, user privacy should not be neglected by search providers. Google has a plethora of online applications such as Gmail, YouTube, Maps, Sheets, Docs, Earth, Drive etc. and all these applications could serve as points to collect user data. To better understand the privacy offered by the most used search engine i.e., Google, comparison with DuckDuckGo is done.

Google states in its privacy policy that it logs all search queries provided by a user along with time, date, browser details, operating system information, IP address and even the search URL. The data collected increases when a user is signed into their Google account. Content created by users such as emails, photos, videos, documents, and spreadsheets that are uploaded or

received through their platform are also collected. They claim this is done to remember language preferences across browsing sessions. Google also passes a portion of this collected data to various affiliates, subsidiaries, and business partners.

DuckDuckGo and Startpage are some of the well-known privacy respecting search engines. Among these, DuckDuckGo has the highest number of users. DuckDuckGo does not collect user data like IP address, cookies and does not share it with other third parties. More importantly, when links opened through Google, Google passes on the search terms to the website in the HTTP header. DuckDuckGo prevents this by anonymizing the search terms using encryption. Additionally, it shows links only to the encrypted version of the sites. When storing user data with an external party such as Google, user data is always at risk of data breach.

Users prefer Google over DuckDuckGo due to its search quality. Comparing strictly relevance of search results, Google is simply better (Hollingsworth, 2019). Therefore, a good alternative for users not willing to compromise on quality but care for privacy is Startpage. Startpage has the best qualities of DuckDuckGo and Google. It is a privacy-respecting search engine that truly values user privacy by not collecting more data than necessary while also displaying quality results similar to Google.

5.3.3 Proxy

Proxy servers provide users the ability to filter and alter traffic by placing itself as an intermediary node between user and the internet. Any incoming and outgoing connection needs to pass through the proxy. This allows for some level of anonymity if configured accordingly. There are also web-based proxy services that provide browser-level protection. There are several web-based proxy services that is usually not recommended to use on a regular. For better security it is suggested to locally install and configure a proxy server. A proxy server can be configured to block advertisements, cookies, and other trackers by filtering user GET requests. Two well-known proxy servers for Linux are Privoxy and Polipo. The main advantage of using proxy server over browser extension ad-blockers are improved load times. However, the initial setup of a proxy server may not be something the regular user would want to do.

```
URLS = (
    "https://easylist-downloads.adblockplus.org/malwaredomains_full.txt"
    "https://easylist-downloads.adblockplus.org/fanboy-social.txt"
    "https://easylist-downloads.adblockplus.org/easyprivacy.txt"
    "https://easylist-downloads.adblockplus.org/easylist.txt"
    "https://easylist-downloads.adblockplus.org/easylistdutch.txt"
)
```

Figure 19 Blocklist Privoxy

Privoxy by default runs on port 8118. To filter content, a blocklist will need to be manually setup. This can be done by adding blocklist URLs to the file. Fig. 19 shows a sample of list that can be added. This can be customised as per user's needs.

5.3.4 Browser Profile: Compartmentalization

To enhance online privacy, it could be beneficial to understand and practice the concept of compartmentalization. As shown in previous sections, leaving an online fingerprint is almost inevitable, and as you browse the web, even with tracking prevention tools, a certain level of your information is exposed. The use of different browsers for different activities will hinder fingerprinting. The basic idea behind compartmentalization is a certain browser is only tied to a few sets of user's interests. Therefore, even with active fingerprinting, trackers get limited data (Segun, 2020).

The limitations of this practice are both related to storage. Firstly, installation of multiple browsers on a device will take up a lot of space. Storage capacity constraints may not allow for this. Secondly, as discussed in section 3.1.1.1.2, some tracker storage mechanisms like LSOs are shared between browsers. This common storage could be used to store identifiers that may uniquely identify user even across different browsers. A potential solution to this problem is suggested in section 5.5

5.4 Additional Tools

5.4.1 Virtual Private Network (VPN)

One common data point collected for fingerprinting purposes is user IP address. It is relatively easy to gather user IP. With the use of networking tools, the geographical location can be derived with obtained IP address. It is also relatively easy to conceal user IP address by either using web-based proxies as discussed in section 5.3.3 or with the use of additional tools such as VPN. The primary function of a VPN is to tunnel user traffic through a private network. Since this connection is encrypted by extending a private network, it makes it difficult for third-party tracking. Third party trackers here mainly include the user's ISP since the DNS requests are encrypted, ISPs lose their ability to view user browsing history. However, HTTP encryption is now provided by most websites by implementing TLS. According to Firefox telemetry data in 2018, 70% of page loads use HTTPS. Simply using a VPN does not drastically improve privacy as claimed by many VPN service providers. With regards to limiting exposure, a VPN merely changes a user IP address. As discussed earlier, IP address is not required for most tracking techniques. Additionally, VPN providers log user activity for troubleshooting. Although most providers make this proprietary information that they promise not to share with anyone, the collection and storage of this data shows that risk is not zero. Data at rest is data at risk. Therefore, one needs to take into consideration the VPN server's geographical location and its log policy. VPNs also affect network quality. A 2019 study conducted by Iqbal et. al. showed VPN connection to negatively impact network quality by increasing packet loss rate from 7.8% to 20.2% and reducing throughput from 82.8% to 71.6% (Iqbal & Riadi, 2019). This section emphasizes the privacy shortcomings of VPN because as widely believed, using a VPN is not the final solution to privacy. Therefore, using VPN is only suggested in combination with some previously mentioned privacy enhancing measures.

A better solution to conceal IP address would be with the use of TOR browser. Features of TOR were compared in an earlier section. TOR provides exceptional anonymity when used right. From the tests performed earlier, it has shown to be the best browser for privacy. It directs the user traffic through the TOR network consisting of multiple nodes. It takes three hops to reach the destination starting from the origin point which is the entry node, followed by the middle relay and finally reaching the exit relay. Improved privacy is, therefore, a well-desired side-effect of this increased complexity. TOR browser, however, is significantly slower for everyday browsing. The average user may simply not want to sacrifice speed for privacy. The use of TOR is only preferred when privacy is most desired.

5.5 General good practices

Most important privacy preserving techniques are discussed in Section 5.1 to Section 5.4. This section discusses a few worthy mentions.

5.5.1 FLoC Opt-out

As discussed earlier in this report, the privacy enhancement promised by Google with FLoC may not be optimal. It is, however, an emerging technology still its infancy and due to the heavy criticism received, FLoC could be different from what was initially proposed. At the time of

writing this report, Google had announced to delay the widespread adoption of FLoC in Chrome browsers by almost a year from the initial release date (Vranica, 2021). Therefore, one may wish to opt-out of FLoC until the final version of technology is out and enough data is available on its real-world usage. Google is currently testing FLoC in Chrome browsers version 89 and above. Some users may be a part of the testing group which can be opted-out by simply using a different browser. Additionally, Chrome users can manually opt out of FLoC by changing the settings.

5.5.2 Browser Storage

If improving privacy is the main priority, it is good practice to periodically clear browser storage. This includes cache, cookies, and browsing history. All modern browsers allow the deletion of cookies such as Flash storage that were not allowed in older versions. But this comes at the cost of user convenience. The relationship between safeguarding privacy and convenience is discussed further in a later section.

5.5.3 Public Email

Different websites and service providers allow users to sign up for their service using email addresses. Some services may also choose to make their user's email address publicly available. When a user is associated with a particular niche service or product, it is possible to learn about their interests and target them with relevant advertisement emails. Therefore, to circumvent this, Email aliases can be used. With the use of aliases, one need not create multiple email accounts that they may find difficult to manage. Instead, multiple aliases could be created forwarding all mails to one account. Some of the well-know email alias providers are Mailnator and Jetable.

6 Privacy Enhancement: Proposed Solution

Based on our understanding of the workings of web trackers, a sound set of guidelines can be developed. One important aspect that we could take into consideration are the challenges faced by tracking companies. A change of perspective can help us devise a better solution. For example, consider the inaccuracies generated during tracking. No tracking technique is one hundred percent accurate, and we know there are ways of over and under estimating statistics. Our proposed solution can capitalise on this. Let us first consider the challenges faced by tracking companies, review our tests performed in Section 5, and design an easy-to-follow set of rules for the general user. There cannot be a single one-size-fits-all solution to this problem as different users may have different user experience expectations. Therefore, our rules shall be divided into three categories: I, II, and III.

Category I: User cares for privacy but does not want to take the extra effort to protect it. This is the average internet user who would want better privacy but does not care too much due to the initial setup. This setup offers the least privacy protection out of the three recommendations but is very convenient to initialise.

Category II: User desires better privacy than the average user and is willing to take some extra steps. This user understands that despite his efforts, tracking is still inevitable but taking precautions greatly reduces his online footprint. There is some effort needed to initially configure the setup but his online activities for the most part remain unaffected. This setup offers better privacy than category I but requires some effort to initially configure.

Category III: User wants best privacy protection. The privacy measures adopted by this user are not always the easiest to setup and use for daily browsing. Privacy tools in this setup will inevitably hinder usage. This setup is usually not sustainable for the average user. This setup, however, does not guarantee absolute anonymity but offers much better privacy in comparison to category I and II.

Recommendations are first presented followed by justification.

Category I

Browser: Edge / Chrome / Firefox / Brave

Justification: All these browsers offer decent private mode with support for the recommended add-ons. At the time of writing this report, HTTPSEverywhere has not been made available in Safari but it has been confirmed to be released in the future (HTTPS Everywhere FAQ, 2016). Therefore, Safari is not included here.

Add-ons: uBlock Origin, optionally HTTPSEverywhere

Justification: From our tests we conclude uBlock origin to be the best performing extension with no additional steps after installation. In terms of ease of use uBlock origin is preferred over other extensions.

Recommended practices:

- To carefully accept desired cookies when presented with cookie consent form.
- Ensure traffic at all times is encrypted wherever possible.
- Periodically delete cookies.
- Opt-out of FLoC

Category II

Browser: Firefox / Brave

Justification: Firefox and privacy-focused Brave offer better privacy features compared to the rest. Chrome is replaced with Brave as they are both built on the same platform and Brave comes with pre-installed privacy features also not at risk of FLoC.

Add-ons: uBlock origin, NoScript

Justification: As shown earlier, NoScript is the best add-on for blocking scripts. But it needs some manual tweaking. NoScript blocks all scripts which may render some websites useless. For this, it is recommended to allow scripts of that domain. This can be done by allowing those scripts to the 'Trusted' list. The order of the scripts listen in NoScript will typically start with scripts of that domain so it is most likely to be the first few scripts. Providing such detailed visibility is one of NoScript's best advantage.

Additional tools: VPN

Justification: To use a VPN that has a privacy respecting log policy. This provides basic IP obfuscation and also encrypts DNS requests. VPN are also useful to bypass internet censorship.

Recommended practices:

- All Category I recommendations.
- Using privacy focused search engines.
- Changing browser settings to send DNT requests and review permissions. Only allow permission to services when necessary.
- The use of email aliases to better protect online identity.
- Compartmentalization at the browser level.

Category III

Browser: TOR browser / Firefox through TOR

Justification: TOR browser provides maximum privacy protection by routing all its traffic through the TOR network. This protects user IP address and increases anonymity. However, TOR offers poor network performance and the slow loading speeds may not be ideal for the average user. Another option would be to run TOR and Privoxy services in combination. This way user can use a browser like Firefox and manually configure proxy settings to be compatible with Privoxy.

Add-ons: Pre-installed Add-ons

Justification: TOR comes bundled with HTTPSEverywhere and NoScript by default. No further add-ons are recommended. TOR project themselves advise against installing any further add-ons. As shown earlier, NoScript in its default state blocks majority of fingerprinting scripts. The use of extra add-ons increases the uniqueness of the browser. As majority of TOR users use it with only default add-ons.

Additional tools: Privoxy

Justification: Privoxy can be provided with rules to avoid downloading certain scripts, banners, images etc. The rulesets available for Privoxy are more comprehensive than lists such as EasyList. Often times EasyList, EasyPrivacy and other notable well-known lists can be combined and are easily available to be run with Privoxy. When combined with TOR it is extremely efficient in protecting privacy.

Recommended practices:

- All Category II recommendations.
- Different online identity. Not revealing real name anywhere online.
- Compartmentalization at the Operating system level. This can be achieved by Whonix or Tails operating systems.
- Not using TOR in Fullscreen mode as that could give away screen resolution hence improving accuracy of the generated fingerprint.

7 Performance Comparison

In the previous section, we compare the three recommended set of configurations to improve privacy. In this section, we evaluate the benefits of each of these configurations to show the difference it would make.

Baseline performance:

With a bare browser we see the crawl through our list detect over 868 third party cookies and 116 fingerprinting scripts. This section attempts to estimate the improvement each configuration could bring to privacy.

Category I

With the recommended browser and uBlock Origin, one can reduce the number of third party significantly. In our test we saw a reduction of around 72% which may differ based on usage but uBlock origin has been proven to be one of the best extensions and is recommended for best privacy protection with no initial setup. We also observe fingerprinting to go down by 44%.

Category II

This configuration is what will work best for most people. It may take some effort to initially set it up. Especially, with adding whitelist domains to NoScript to make sites function right. NoScript by default blocks more than 90% of scripts which is good for privacy but severely affects usability. At the bare minimum, this configuration guarantees better protection than Category I. From our results, we see NoScript to block 93% of scripts but that will reduce in real-world usage.

Category III

The initial setup as well as the overall user experience in this configuration may simply not be desirable for everyday usage. TOR network offers exceptional anonymity and configuring privoxy with the right set of lists will block content that NoScript did not. When this combined with compartmentalization at the OS level, fingerprinting results will also be inaccurate. However, no setup can make a user completely anonymous. It is simply adding layers of protection to make tracking difficult.

8 Conclusion

It is important to understand that achieving optimal privacy is something that needs consistent effort. Enhancing privacy should be seen as a process. From the discussion and comparison of various privacy improving tools, one can observe that privacy often comes at the cost of convenience. Therefore, to protect a user's privacy, the user will need to develop good habits that maintain and improve their current level of privacy. As shown above in some of the demonstrations, a small mistake could compromise user identity. The typical user may not desire that level of anonymity but it shows us the advancement of tracking tools. We have witnessed the level and complexity of tracking increase over the years. This report only covers the most prominent techniques. Advertisement companies are heavily dependent on web tracking technologies and there will be newer techniques which will introduce newer problems. Hence, one must always stay prepared to adopt necessary measures when required.

8.1 Achievements

This report presents a detailed study about the current state of the most prominent tracking mechanisms. The first part of the report uncovers the techniques used in great detail in hopes of educating the user. By understanding the technology, we are at a better position to tackle them and able to make sound decisions when in doubt. The second part of the report presents a detailed study on the preventive measures one can take. The proposed solution has been put together based on numerous other studies. The achievement of this report is that it presents the reader a well-tested set of guidelines that will improve privacy.

8.2 Future Work

In this age of information, privacy is at a greater risk than ever before. As technology evolves newer threats emerge. Future work regarding tracking will involve studies of new upcoming technologies that are either not widely adopted yet or lack access to obtain meaningful data.

9 Critical Appraisal

Initially the research plan of this report was to study the evolution and functioning of web trackers. For this the chosen domains were Google, YouTube, Amazon, and Facebook. These were our chosen domains since they were the top domains according to Alexa generated list. This presented us with two problems: Firstly, the chosen domains were too few to obtain information about third party tracking so as the project progressed, the crawl list was increased to the top 50 domains to gain a better understanding of the distribution of cookies. This deviation from the plan removed the focus from the proposed domains of research but resulted in a better quality research. Secondly, upon investigating web trackers, there were various other kinds of web tracking that are either used across devices other than desktops or not very commonly found. Tracking techniques like geofencing, ultrasound audio beacons, Wi-Fi triangulation etc. are some examples of tracking that were found later on. These tracking techniques were utilizing complex techniques that do not involve web browsers. They depend on the internet however as mentioned in the earlier section of the report, this study is limited to desktop web browsers. Also, due to the limited access of FLoC during writing this report, practical analysis could not be performed as extensive as done for other techniques. Moreover, there is no guarantee that the final FLoC functioning will be exactly as proposed. Lastly, we try our best to isolate the environment and conduct tests in a controlled virtual machine. However, tracking information that is seen by tracking companies was challenging to obtain and apart from the auditing tools used, we do not gain insight about what trackers actually see.

Bibliography and References

10 Ad Blocking Extensions Tested for Best Performance. (2015, August 26). Raymond.CC Blog. <https://www.raymond.cc/blog/10-ad-blocking-extensions-tested-for-best-performance/view-all/>

Acar, G., Eubank, C., Englehardt, S., Juarez, M., Narayanan, A., & Diaz, C. (2014). The Web Never Forgets. Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security - CCS '14. <https://doi.org/10.1145/2660267.2660347>

Agarwal, G., Bursztein, E., Jackson, C., & Boneh, D. (2010, September). An Analysis of Private Browsing Modes in Modern Browsers. ResearchGate. https://www.researchgate.net/publication/221260449_An_Analysis_of_Private_Browsing_Modes_in_Modern_Browsers

Anderton, K. (2020, December 13). Middle Schooler Proves Google Search Results Influence Political Opinions [Infographic]. Forbes. <https://www.forbes.com/sites/kevinanderton/2020/12/13/middle-schooler-proves-google-search-results-influence-political-opinions-infographic/>

Boda, K., Földes, Á. M., Gulyás, G. G., & Imre, S. (2012). User Tracking on the Web via Cross-Browser Fingerprinting. Information Security Technology for Applications, 31–46. https://doi.org/10.1007/978-3-642-29615-4_4

Bohn, D. (2021, June 24). Google delays blocking third-party cookies in Chrome until 2023. The Verge. <https://www.theverge.com/2021/6/24/22547339/google-chrome-cookiepocalypse-delayed-2023>

Bridges, J. (2011, September 27). How to protect your online data from insurance companies. ReputationDefender. <https://www.reputationdefender.com/blog/privacy/how-protect-your-online-data-insurance-companies>

Bujlow, T., Carela-Espanol, V., Lee, B.-R., & Barlet-Ros, P. (2017). A Survey on Web Tracking: Mechanisms, Implications, and Defenses. Proceedings of the IEEE, 105(8), 1476–1510. <https://doi.org/10.1109/jproc.2016.2637878>

Derksen, I. (2016, July 7). HTML5 Tracking Techniques inPractice. Radboud University; Radboud University. https://www.cs.ru.nl/bachelors-theses/2016/lvar_Derksen___4375408___HTML5_Tracking_Techniques_in_Practice.pdf

Eckersley, P. (2010). How Unique Is Your Web Browser? In Privacy Enhancing Technologies (pp. 1–18). https://doi.org/10.1007/978-3-642-14527-8_1

Gomer, R., Rodrigues, E. M., Milic-Frayling, N., & Schraefel, M. C. (2013). Network Analysis of Third Party Tracking: User Exposure to Tracking Cookies through Search. 2013 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT). <https://doi.org/10.1109/wi-iat.2013.77>

Hollingsworth, S. (2019, April 12). DuckDuckGo vs. Google: An In-Depth Search Engine Comparison. Search Engine Journal. <https://www.searchenginejournal.com/google-vs-duckduckgo/301997/>

Howell O'Neill, P. (2015, April 16). "Geo-inference" can reveal your location in all mainstream browsers. The Daily Dot. <https://www.dailydot.com/unclick/geo-inference-attack-google-craigslist-maps/>

HTTPS Everywhere FAQ. (2016, November 7). Electronic Frontier Foundation. <https://www.eff.org/https-everywhere/faq#will-there-be-a-version-of-https-everywhere-for-ie-safari-or-some-other-browser>

Iqbal, M., & Riadi, I. (2019). Analysis of Security Virtual Private Network (VPN) Using OpenVPN. International Journal of Cyber-Security and Digital Forensics (IJCSDF), 8(1), 58–65. <https://sdiwc.net/digital-library/analysis-of-security-virtual-private-network-vpn-using-openvpn>

Jia, Y., Dong, X., Liang, Z., & Saxena, P. (2015). I Know Where You've Been: Geo-Inference Attacks via the Browser Cache. IEEE Internet Computing, 19(1), 44–53. <https://doi.org/10.1109/mic.2014.103>

Li, T.-C., Hang, H., Faloutsos, M., & Efstathiopoulos, P. (2015). TrackAdvisor: Taking Back Browsing Privacy from Third-Party Trackers. Passive and Active Measurement, 277–289. https://doi.org/10.1007/978-3-319-15509-8_21

Lobosco, K. (2013, August 26). Facebook friends could change your credit score. CNNMoney. <https://money.cnn.com/2013/08/26/technology/social/facebook-credit-score/index.html>

M. McDonald, A., & Faith Cranor, L. (2011, January 31). A Survey of the Use of Adobe Flash Local Shared Objects to Respawn HTTP Cookies. Research Gate. https://www.researchgate.net/publication/228566536_A_survey_of_the_use_of_Adobe_Flash_Local_Shared_Objects_to_respawn_HTTP_cookies

Mayer, J. (2013, December 13). How the NSA Piggy-Backs on Third-Party Trackers. Cyberlaw.stanford.edu. <http://cyberlaw.stanford.edu/publications/how-nsa-piggy-backs-third-party-trackers>

Mayer, J. R., & Mitchell, J. C. (2012). Third-Party Web Tracking: Policy and Technology. 2012 IEEE Symposium on Security and Privacy, 12851630. <https://doi.org/10.1109/sp.2012.47>

Mohamed, N. (2009, October 8). You Deleted Your Cookies? Think Again. Wired. <https://www.wired.com/2009/08/you-deleted-your-cookies-think-again/>

Mowery, K., & Shacham, H. (2012). Pixel Perfect : Fingerprinting Canvas in HTML 5. Www.semanticscholar.org. <https://www.semanticscholar.org/paper/Pixel-Perfect-%3A-Fingerprinting-Canvas-in-HTML-5-Mowery-Shacham/9243f5103669bd0e5b29f333f6e1d2246dc0a492>

Nikiforakis, N., Kapravelos, A., Joosen, W., Kruegel, C., Piessens, F., & Vigna, G. (2013). Cookieless Monster: Exploring the Ecosystem of Web-Based Device Fingerprinting. 2013 IEEE Symposium on Security and Privacy. <https://doi.org/10.1109/sp.2013.43>

Nouwens, M., Liccardi, I., Veale, M., Karger, D., & Kagal, L. (2020). Dark Patterns after the GDPR: Scraping Consent Pop-ups and Demonstrating their Influence. <https://doi.org/10.1145/3313831.3376321>

Quintel, D., & Wilson, R. (2020). Analytics and Privacy: Information Technology and Libraries, 39(3). <https://doi.org/10.6017/ital.v39i3.12219>

Ravichandran, D., & Vassilvitskii, S. (2021). Evaluation of Cohort Algorithms for the FLoC API. Mozilla.

<https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&cad=rja&uact=8&ved=2ahUKEwj4MGz6pzyAhVTZxUIHRh5CxiQFnoECAIQAw&url=https%3A%2F%2Fraw.githubusercontent.com%2Fgoogle%2Fads-privacy%2Fmaster%2Fproposals%2FFLoC%2FFLOC-Whitepaper-Google.pdf&usg=AOvVaw1UXgprEuwDQrSa4J509wAQ>

Rescorla, E., & Thomson, M. (2021). Technical Comments on FLoC Privacy. Google.com.

https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&cad=rja&uact=8&ved=2ahUKEwidtKDX6ZzyAhWjuXEKHfdUCXgQFnoECAIQAw&url=https%3A%2F%2Fmozilla.github.io%2Fppa-docs%2Ffloc_report.pdf&usg=AOvVaw0QhwCzNCZ2vMXIkXzt0Rn0

Roesner, F., Kohno, T., & Wetherall, D. (2012). Detecting and Defending Against Third-Party Tracking on the Web. Wwww.usenix.org. <https://www.usenix.org/conference/nsdi12/technical-sessions/presentation/roesner>

Segun -, D. (2020, April 25). Browser Compartmentalization: How to Compartmentalize Your Web Browsers. SecureBlitz Cybersecurity. <https://secureblitz.com/browser-compartmentalization/>

Unger, T., Mulazzani, M., Frühwirth, D., Huber, M., Schrittwieser, S., & Weippl, E. (2013, September 1). SHPF: Enhancing HTTP(S) Session Security with Browser Fingerprinting. IEEE Xplore. <https://doi.org/10.1109/ARES.2013.33>

Urton, J. (2016, August 15). Computer scientists reveal history of third-party web tracking. ScienceDaily. <https://www.sciencedaily.com/releases/2016/08/160815111429.htm>

Utz, C., Degeling, M., Fahl, S., Schaub, F., & Holz, T. (2019). (Un)informed Consent: Studying GDPR Consent Notices in the Field ACM Reference Format. <https://doi.org/10.1145/3319535.3354212>

Vranica, P. H., Sam Schechner and Suzanne. (2021, June 24). Google Delays Cookie Removal to Late 2023. Wall Street Journal. <https://www.wsj.com/articles/google-delays-cookie-removal-to-late-2023-11624542064>

w3techs. (2021). Usage Statistics and Market Share of Google Analytics for Websites, January 2021. W3techs.com. <https://w3techs.com/technologies/details/ta-googleanalytics>

Weinberg, G. (2021). What are the biggest tracker networks and what can I do about them? SpreadPrivacy. <https://spreadprivacy.com/biggest-tracker-networks/>

Weinberger, A. (2011). The Impact of Cookie Deletion on Site-Server and Ad-Server Metrics in Australia. ComScore. http://www.comscore.com/content/download/7251/125689/file/Impact+of+Cookie+Deletion+Australia_January+2011.pdf

Appendix A – Requirements Specification Document

10 Virtual Machine specifications

Ubuntu 64-bit

Operating System: Ubuntu 20.10

OS Type: 64-bit

GNOME version: 3.38.1

Windowing System: X11

Virtualization: VMware

Device name: ubuntu

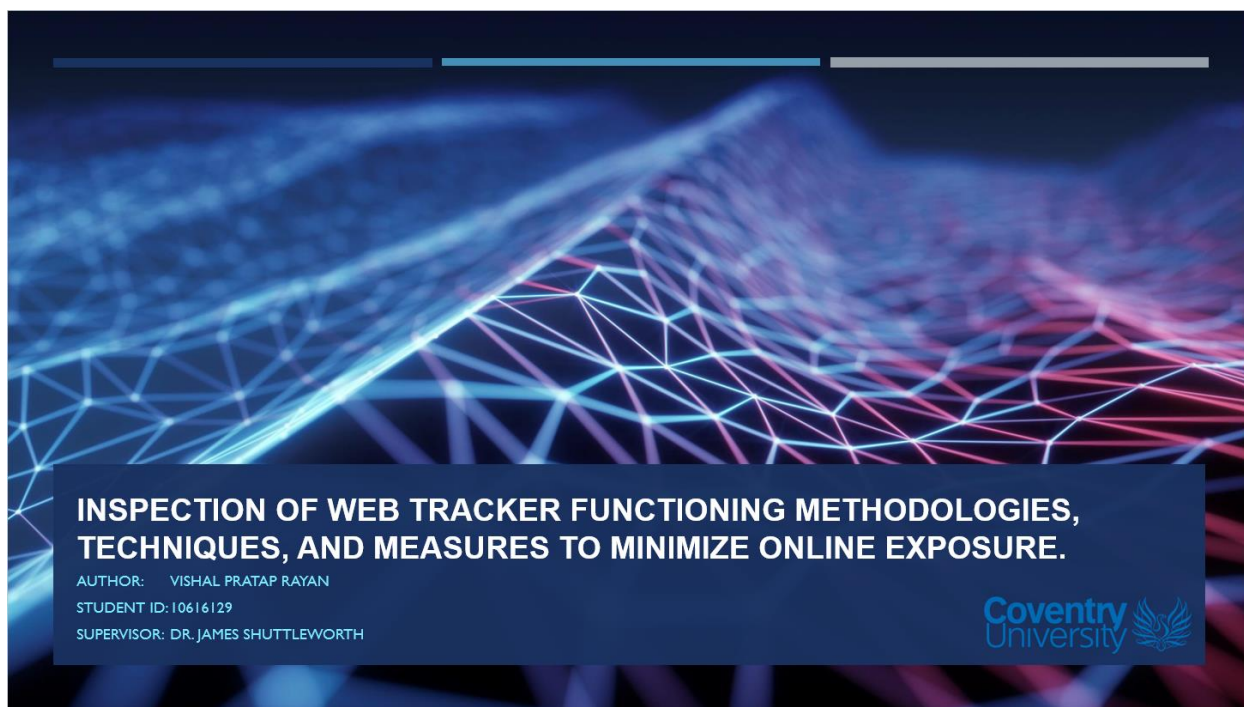
Memory: 3.8 GB

Processor: Intel Core i7-10750H CPU @ 2.6

Graphics: SVGA3D; build: RELEASE; LLVM;

Disk capacity: 26.8 GB

Appendix B – Project Presentation

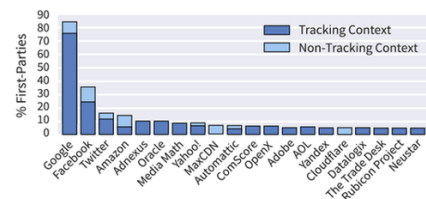


OBJECTIVES

- 1. Analyze current state of web trackers to obtain better understanding.
- 2. Propose a solution to minimize excessive online exposure.
 - Three types of recommended approaches

BACKGROUND

- Broad topic area: Online privacy
- Narrow topic area: Web trackers
- Online privacy and its importance.
- Web trackers:
 - What are they? (different types)
 - Why are they used? (pros and cons, different use cases)
 - How do they function? (explanation and analysis)
 - Future of web tracking? (upcoming technologies)



source: spreadprivacy.com

BACKGROUND

- Web Trackers use cases:
 - 1. Advertising
 - 2. Analytics
 - 3. Price discrimination
 - 4. Credit score assessment
 - 5. Biased Search
 - 6. Surveillance
 - 7. Insurance and Background check

BACKGROUND (CONT.)

- Effective Measure against tracking:
 - Browsers
 - Browser configuration
 - Extensions
 - Other helpful tools and practices

LITERATURE REVIEW

- Third Party Tracking
 - Third party Web-Tracking: Policy and Technology, idea of web measurement, FourthParty [1]
- Evercookie
 - In 2014, Persistent web tracking, Acar et. al [2]
- Fingerprinting
 - Boda et. al introduces browser fingerprinting [3] .Shanon entropy a useful measure to calculate uniqueness.
- FLoC
 - Initial stages of development, FLoC whitepaper [4]and Mozilla privacy report [5]
- Privacy enhancements

LITERATURE REVIEW (CONT.)

- Outdated studies that do not account for latest developments in tracking technology.
 - FLoC.
 - Recommendations made have been tested recently.



METHODOLOGY

PART - I

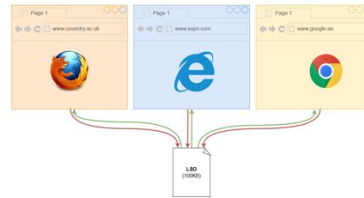
- Web Trackers:
 - Storage-based tracking - 2
 - Cache-based tracking
 - Fingerprinting-based tracking - 6
 - Future of web trackers?



METHODOLOGY

PART - I

- Storage-based techniques – Function and Findings
 - HTTP cookie
 - Flash cookie
- Cache-based technique
 - E-Tag



METHODOLOGY

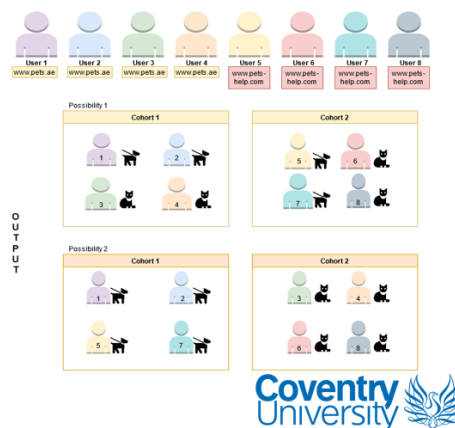
PART - I

- Fingerprinting-based techniques
 - Device
 - Network
 - Battery Status API
 - Audio
 - Browser
 - Canvas
 - Operating System

METHODOLOGY

PART - I

- Future of web tracking?
 - Federated Learning of Cohorts (FLoC)



METHODOLOGY

PART - II

- Browser choice – Chrome, Firefox, Edge, TOR, Safari
 - Alternate consideration: Brave?
- Browser configuration
 - Execution blockers - [NoScript](#)
 - Third-party tracker blockers – [Adblock plus](#), [Ghostery](#), [uBlock Origin](#), [Privacy Badger](#), [WebOfTrust](#)

METHODOLOGY

PART - II

- Browsing behavior
 - Cookie consent
 - Search engine preference – DuckDuckGo, Startpage
- Proxy
 - Privoxy, Polipo, web-based
- Compartmentalization



METHODOLOGY

PART - II

- Additional Tools
 - VPN, why or why not TOR instead?
- General good practices
 - Browser storage
 - Email alias
 - FLoC opt-out



METHODOLOGY

PART - II

- Privacy enhancement measures comparison
- Privacy vs Convenience relationship



RESEARCH RESULTS

- Three recommended set of configurations:
- I. Convenient setup, better privacy than bare browser
- II. Improved privacy, some initial setup (best balance)
- III. Maximum privacy, inconvenient regular usage



REFERENCES

- [1] Mayer, J. R., & Mitchell, J. C. (2012). Third-Party Web Tracking: Policy and Technology. *2012 IEEE Symposium on Security and Privacy*, 1285–1300. <https://doi.org/10.1109/sp.2012.47>
- [2] Acar, G., Eubank, C., Englehardt, S., Juarez, M., Narayanan, A., & Diaz, C. (2014). The Web Never Forgets. *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security - CCS '14*. <https://doi.org/10.1145/2660267.2660347>
- [3] Boda, K., Földes, Á. M., Gulyás, G. G., & Imre, S. (2012). User Tracking on the Web via Cross-Browser Fingerprinting. *Information Security Technology for Applications*, 31–46. https://doi.org/10.1007/978-3-642-29615-4_4
- [4] Rescorla, E., & Thomson, M. (2021). *Technical Comments on FLoC Privacy*. Google.com. https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&cad=rja&uact=8&ved=2ahUKewidKDX6ZzyAhWjuXEKHfdUCXgQFnoECAIQAw&url=https%3A%2F%2Fmozilla.github.io%2Fpfp-docs%2Ffloc_report.pdf&usg=AOvVaw0QhwCzNCZ2vMXlkXzt0Rn0
- [5] Ravichandran, D., & Vassilvitskii, S. (2021). *Evaluation of Cohort Algorithms for the FLoC API*. Mozilla. <https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&cad=rja&uact=8&ved=2ahUKewjm4MGz6pzyAhVTZxUIHRh5CxlQFnoECAIQAw&url=https%3A%2F%2Fraw.githubusercontent.com%2Fgoogle%2Fads-privacy%2Fmaster%2Fproposals%2FFLoC%2FFLOC-Whitepaper-Google.pdf&usg=AOvVaw1UXgprEuWDQrSa4j509wAQ>



THANK YOU

- VISHAL PRATAP RAYAN



Appendix C – Certificate of Ethics Approval

Inspection of Web tracker functioning methodologies, techniques and measures to minimize online exposure.

P122719



Certificate of Ethical Approval

Applicant: Vishal Rayan
Project Title: Inspection of Web tracker functioning methodologies, techniques and measures to minimize online exposure.

This is to certify that the above named applicant has completed the Coventry University Ethical Approval process and their project has been confirmed and approved as Low Risk

Date of approval: 07 Jun 2021
Project Reference Number: P122719

Appendix X

















Abbreviations

HTTP	Hyper-text Transfer Protocol
PII	Personally Identifiable Information
DNT	Do Not Track
CSS	Cascading Style Sheet
HTML	Hyper-text Markup Language
HTML5	Hyper-text Markup Language version 5
LSO	Local Shared Objects
OS	Operating System
FLoC	Federated Learning of Cohorts
W3C	World Wide Web Consortium
API	Application Programming Interface
GPU	Graphics Processing Unit
TOR	The Onion Router
VM	Virtual Machine
WRTC	Web Real-time Communication
EFF	Electronic Frontier Foundation
ISP	Internet Service Provider

PHP Snippet to get user IP

```
if (!empty($_SERVER['HTTP_CLIENT_IP'])) {  
    $ip = $_SERVER['HTTP_CLIENT_IP'];  
} elseif (!empty($_SERVER['HTTP_X_FORWARDED_FOR'])) {  
    $ip = $_SERVER['HTTP_X_FORWARDED_FOR'];  
} else {  
    $ip = $_SERVER['REMOTE_ADDR'];  
}
```

AudioContext Supported Browsers

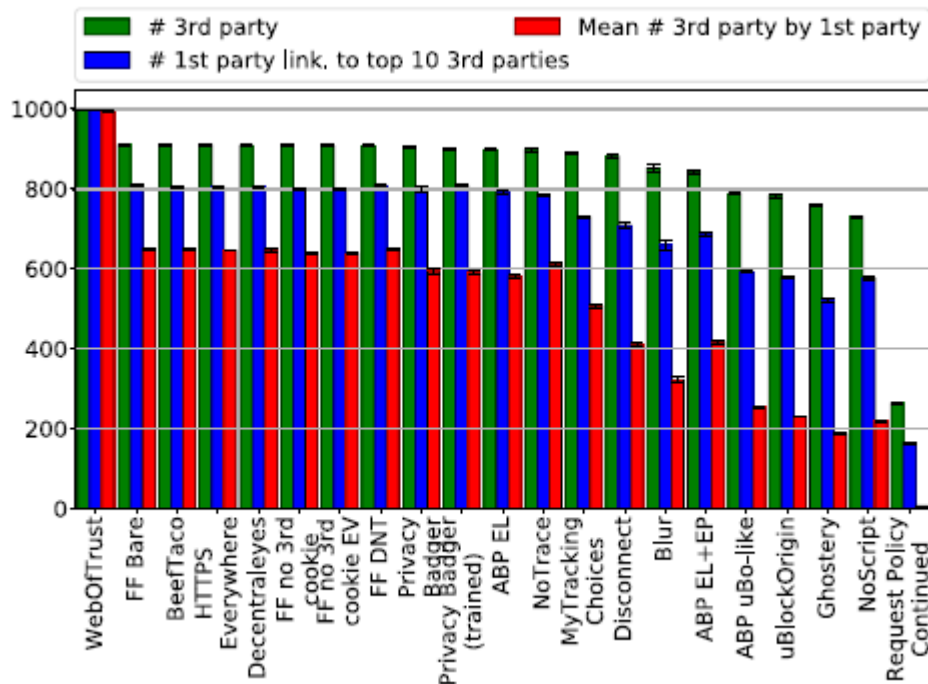
												
	Chrome 	Edge 	Firefox 	Internet Explorer 	Opera 	Safari 	WebView Android 	Chrome Android 	Firefox for Android 	Opera Android 	Safari on iOS 	Samsung Internet 
AudioContext	35 ▼	12	25	No	22 ▼	14.1 ▼	37 ▼	35 ▼	25	22 ▼	14.5 ▼	3.0 ▼
AudioContext() constructor	35 ★ ▼	12	25	No	22 ★ ▼	14.1 ▼	37 ★ ▼	35 ★ ▼	25	22 ★ ▼	14.5 ▼	3.0 ★ ▼
options.latencyHint parameter	60	79	61	No	47	No ★ ▼	60	60	61	44	No ★ ▼	8.0
options.sampleRate parameter	74	79	61	No	62	14.1	74	74	61	53	14.5	11.0
baseLatency 	58	79	70	No	45	14.1	58	58	No	43	14.5	7.0
close	42	14	40	No	Yes	9	43	43	40	Yes	9	4.0
createMediaElementSource	14	12	25	No	15	6	37	18	25	14	6	1.0
createMediaStreamDestination	14	79	25	No	15	6	37	18	25	14	Yes	1.0
createMediaStreamSource	14	12	25	No	15	6	37	18	25	14	Yes	1.0
createMediaStreamTrackSource	No	No	68 ★ ▼	No	No	No	No	No	68 ★ ▼	No	No	No
getOutputTimestamp 	57	79	70	No	44	14.1	57	57	No	43	14.5	7.0
outputLatency	No	No	70	No	No	No	No	No	No	No	No	No
resume	41	14	40	No	Yes	9	41	41	Yes	Yes	9	4.0
suspend	43	14	40	No	Yes	9	43	43	40	Yes	9	4.0

Usage Share of Desktop Browsers

Browser	StatCounter ^[18] June 2021	NetMarketShare ^[19] May 2021	W3Counter ^[20] November 2019	Wikimedia ^[21] August 2020
Chrome	68.76%	69.57%	59.3%	54.9%
Edge	8.39%	12.16%	4.2%	6.1%
Safari	9.70%	3.45%	14.6%	9.4%
Firefox	7.17%	6.26%	6.1%	13.3%
Opera	2.47%	1.09%	3.5%	1.6%
IE	1.45%	3.35%	5.3%	3.7%
Others	3.51%	4.12%	7.0%	11.0%

Source: Wikipedia.com

Privacy Footprint



Source: A comparison of web privacy protection techniques (Mazel et.al,2019)

JavaScript to generate Canvas Fingerprint

```

<b>Hash:</b> <span id='hash-code'></span><br>
<canvas id='myCanvas' width='200' height='40' style='border:1px solid
#000000;'></canvas>
<script>
var canvas = document.getElementById("myCanvas");
var ctx = canvas.getContext("2d");

ctx.fillStyle = "rgb(255,0,255)";
ctx.beginPath();
ctx.rect(20, 20, 150, 100);
ctx.fill();
ctx.stroke();
ctx.closePath();
ctx.beginPath();
ctx.fillStyle = "rgb(0,255,255)";
ctx.arc(50, 50, 50, 0, Math.PI * 2, true);
ctx.fill();
ctx.stroke();
ctx.closePath();

txt = 'Cov#$$^@féú';
ctx.textBaseline = "top";
ctx.font = '17px "Arial 17"';
ctx.textBaseline = "alphabetic";
ctx.fillStyle = "rgb(255,5,5)";
ctx.rotate(.03);
ctx.fillText(txt, 4, 17);
ctx.fillStyle = "rgb(155,255,5)";
ctx.shadowBlur=8;
ctx.shadowColor="red";
ctx.fillRect(20,12,100,5);

src = canvas.toDataURL();
hash = 0;

```

```

for (i = 0; i < src.length; i++) {
  char = src.charCodeAt(i);
  hash = ((hash<<5)-hash)+char;
  hash = hash & hash;
}

$('#hash-code').html(hash);

</script>

```

AudioContext fingerprinting Sample

AudioContext properties:

```

{
  "ac-baseLatency": 0,
  "ac-outputLatency": 0,
  "ac-sampleRate": 48000,
  "ac-state": "suspended",
  "ac-maxChannelCount": 2,
  "ac-numberOfInputs": 1,
  "ac-numberOfOutputs": 0,
  "ac-channelCount": 2,
  "ac-channelCountMode": "explicit",
  "ac-channelInterpretation": "speakers",
  "an-fftSize": 2048,
  "an-frequencyBinCount": 1024,
  "an-minDecibels": -100,
  "an-maxDecibels": -30,
  "an-smoothingTimeConstant": 0.8,
  "an-numberOfInputs": 1,
  "an-numberOfOutputs": 1,
  "an-channelCount": 2,
  "an-channelCountMode": "max",
  "an-channelInterpretation": "speakers"
}

```

Fingerprint using DynamicsCompressor (sum of buffer values):

35.7383295930922

Fingerprint using DynamicsCompressor (hash of full buffer):

2dc43feaa1474319db71be0f4a9810c4a2a54524

Fingerprint using OscillatorNode:

```

-122.10106658935547,-122.17668151855469,-121.51886749267578,-
120.64137268066406,-119.46350860595703,-118.02296447753906,-
116.34650421142578,-114.44546508789062,-112.31880950927734,-
109.9481201171875,-107.31201171875,-104.37184143066406,-
101.08609008789062,-97.42935943603516,-93.51634979248047,-
90.6439208984375,-82.70191192626953,-44.70573425292969,-
31.952308654785156,-29.518218994140625,-36.19671630859375,-
56.003013610839844,-95.44043731689453,-91.57635498046875,-
95.32821655273438,-99.18425750732422,-102.70282745361328,-
105.87161254882812,-108.73149108886719,-111.33251190185547
...

```

Fingerprint using hybrid of OscillatorNode/DynamicsCompressor method:

```

-129.0526123046875,-119.44966125488281,-108.23187255859375,-
101.38246154785156,-102.75074768066406,-114.57632446289062,-
109.89717102050781,-98.22866821289062,-95.67564392089844,-

```

103.14138793945312,-117.47007751464844,-111.5203857421875,-
101.82391357421875,-101.04198455810547,-89.2830810546875,-
93.03608703613281,-89.74244689941406,-53.49012756347656,-
40.75892639160156,-38.325679779052734,-45.00003433227539,-
64.69832611083984,-95.99256896972656,-88.3079833984375,-
94.27742767333984,-98.67274475097656,-107.62666320800781,-
114.7966079711914,-106.75559997558594,-95.29183959960938

Syntax to check CSS Support

```
CSS.supports(propertyName, value);  
CSS.supports(supportCondition);
```

Crawl list

1. Google.com
2. Youtube.com
3. Amazon.co.uk
4. Reddit.com
5. Google.co.uk
6. Bbc.co.uk
7. Bongacams.com
8. Live.com
9. Wikipedia.org
10. Twitch.tv
11. Ebay.co.uk
12. Yahoo.com
13. Facebook.com
14. Chaturbate.com
15. Netflix.com
16. Microsoftonline.com
17. Ladbible.com
18. Www.gov.uk
19. Livejasmin.com
20. Zoom.us
21. Pornhub.com
22. Fandom.com
23. Rightmove.co.uk
24. Theguardian.com
25. Itv.com
26. Aparat.com
27. Trustpilot.com
28. Xhamster.com
29. Vk.com
30. Roblox.com
31. Microsoft.com
32. Imgur.com
33. Sportbible.com
34. Bing.com
35. Dailymail.co.uk
36. Unilad.co.uk
37. Virginmedia.com
38. Lloydsbank.co.uk
39. Redd.it

40. Ok.ru
41. Varzesh3.com
42. Chess.com
43. Msn.com
44. Amazon.com
45. Bt.com
46. Hotukdeals.com
47. Www.nhs.uk
48. Imdb.com
49. Instagram.com
50. Autotrader.co.uk

Crawl Reports – Github links

https://github.com/vishalprataprayan/vishalprataprayan/blob/7f21040c089123d0ec9f90ff62a46f7ec72bfc67/CrawlReport_Main.html

Crawl report: Ghostery

<https://github.com/vishalprataprayan/privacyreportcrawl/blob/main/Ghostery.html>

Crawl report: Adblock plus

<https://github.com/vishalprataprayan/privacyreportcrawl/blob/main/ADblockplus.html>

Crawl report: uBlock Origin

<https://github.com/vishalprataprayan/privacyreportcrawl/blob/main/ublock%20origin.html>

Crawl reports: Privacy Badger

<https://github.com/vishalprataprayan/privacyreportcrawl/blob/main/Privacybadger.html>

https://github.com/vishalprataprayan/privacyreportcrawl/blob/main/Privacybadger_2.html

https://github.com/vishalprataprayan/privacyreportcrawl/blob/main/Privacybader_3.html

https://github.com/vishalprataprayan/privacyreportcrawl/blob/main/Privacybadger_4.html