An Overview of PrivacyTests.org

Clément Gindrier[1], Coraline Mori[1]

Abstract

The browser is an important factor to consider when protecting the users' privacy, since websites collect users' activities on the web and many users use the browsers' default configurations without extensions. `PrivacyTests.org` is an open-source project created by Arthur Edelstein to compare the protection of privacy in different browsers. The test results are published on the website, each table corresponding to a category and containing its privacy tests' results.

In this article, we analyzed the project, the tests, and the results (which we replicated thanks to GitHub sources), and finally, we discussed the state of the art of reviews that showed possible biases.

1. Introduction

As our privacy on the Internet is challenged, we may wonder more than ever how to protect our privacy while surfing on the web. Many of the possibilities to increase our privacy depend on us and how we manage our online life. However, a lot of information can be exchanged with the browser without our knowledge, enabling the websites to profile and track their users. With the goal of comparing how different browsers protect our privacy online, privacy Engineer Arthur Edelstein created the website `PrivacyTests.org`[1].

Arthur Edelstein is a developer engineer, specializing in browser privacy, who worked on Tor for four years, on Firefox for nearly three years and he currently works at Brave since 2022[2]. He also talk a lot about privacy on social network like LinkedIn and made a video about privacy on Tor[3]. In short, Arthur Edelstein is a committed enthusiast and an important figure in the field of browser privacy.

For the project, Arthur Edelstein regularly tests the most commonly used browsers with a chosen set of tests regarding privacy protection and publishes the results on the website, creating a concise way of comparing privacy between different browsers. This list of privacy protection items can also be useful for browsers, which can adapt themselves to try to check all the boxes.

The tests are run approximately every two weeks[4] and the website is updated at that frequency with the obtained results (passed, failed, or no such feature). The tests' source code, as well as the results can be found in the git repository for the project[5]. The browsers are separated into six categories, and each browser is tested on a number of items related to privacy divided into categories.

YouTube's videos and forums have talked about privacytests, explaining how it works and some of its specifications, but we have not found an article detailing all the items or analyzing the website. There were also opinions on forums and websites where browser developers exchanged with Edelstein on the way the browsers were tested and the possible biases. In this article, we will try to categorize and list the different opinions on the topic.

In this article, we will try to understand how reliable and relevant the results are to compare the different browsers. To do that, we first interested ourselves with each of the items presented, then we studied the different reviews on the websites and the possible biases, and finally we tried to replicate the result.

2. Understanding the Website and Tests

The first step in this research was to try to understand everything on the website. We started by listing all the browser categories, as well as the browser versions tested (see annex Browsers). Then, we tried to understand the item categories and made an exhaustive list of all the items, while noting the difference in items tested according to the browser categories (see annex Item Categories). Finally, we detailed all the items and tried to understand how

[1]Grenoble INP - Ensimag, France

useful each of them was (see annex Items Detailed). All of this information is detailed in the annex, however some points are interesting to note.

First, we need to keep in mind that these items test how browsers protect the user against tracking from the websites, but they do not take into account the tracking (like telemetry) from the browsers themselves. The browsers' respect for the users' privacy would be hard to test automatically. This information could be found manually, but even then, it would be hard to know to what extent privacy is respected unless the browser is open source.

We can also note that some categories have many items that are similar. For example, in the last category, all tests work the same but for a different link. If we look at the number of tests that passed (see annex Evolution of Test Results), the number can increase a lot by adding these tests, which could be misleading. That is also the case for the penultimate category. However, since they are both at the end, they are also less prioritized on the website.

Some items do not have the same importance depending on the browser. For instance, LibreWolf recommended not to use his browser with the tor network [6], so even if the test passed, it may be debatable to use it. And so a test that passes or fails doesn't make much difference in this case.

In the test result, it is surprising that Tor fails many tests even though it is considered to be the browser that protects the users' privacy the most. We ask Arthur Edelstein if this is a big deal since we are browsing through VPNs. He answer that "Those two categories are examples of tracking that happens in the browser, even if you are using the Tor network. Tor has very strong privacy protections, but so far it does not block tracking in those two categories. Blocking query parameters would definitely improve Tor Browser privacy. Blocking tracker content is also likely to provide "defense in depth" for cases when a script finds a way to work around an existing Tor Browser privacy protection."

In the categories, the tests are surely chosen with the most possible relevance, it is nevertheless impossible to be exhaustive and impossible to choose witch tests are the most importants. For instance in fingerprinting, he asked his community on a GitHub Issue for ideas [7], this is a good thing because this diversify the ideas. And then he selected a few and completed the list. But different browsers seemed to have different list of fingerprinting mechanisms to protect against, which are not necessarily compatible with the tests selected. However, this is not a big deal, because the goal is to have a general trend. Not to do a very precise and exhaustive analysis.

It is nevertheless interesting to compare the test results to the browsers' claims. For example, Firefox does not pass any of the fingerprinting tests, even though they claim on their website that they block them [8] [9]. To block them, extensions can be used, but that means the claim is inexact. In fact, Firefox does block some fingerprinting, but it is very rare. The website `PrivacyTests.org` helps make this kind of observation.

Finally, it is possible to observe the evolution of the success of the tests by the browsers thanks to the Wayback machine [10], to see which ones have made the most effort. This also can show the impact of the website. Indeed, in one year in 2022, the proportion of tests passed has increased up to 70%. The statistics are detailed in this section 8 of the annexe. As noted in the section1, At the time of writing (January 2023), the evolution of browsers is rather constant, except those of Vivaldi and Opera which are very similar and chaotic. We also notice that Librewolf is still in the lead, but Brave is progressing faster and catching up. And in 3rd place we find Tor.

3. Site biases and reviews: State of the art

The reviews are important because `PrivacyTests.org` can influence the users, and therefore the manufacturers who will follow the demand, and try to maximize the number of passed tests on the website (we can see the improvement of websites since the creation in 2021 [10], As previously stated, there has been a 70% increase in green crosses in the year 2022 2). This allows, in addition to advising people, to improve the privacy of browsers in the long run. However that is only the case if the tests are reliable, relevant, and unbiased. This is what we will see in this section.

First of all, although there are already some browser comparison articles on the internet[11], Arthur Edelstein and us only found `PrivacyTests.org` for the moment which compares browsers with automated tests. This allows for a theoretically less biased, verifiable and more detailed result.

Nevertheless, there are some criticisms that often appear on the Internet. Among the most important:

— It turns out that the tests are performed with the default browser settings (see figure 1). It is a deliberate choice made and defended by the website's creator[12] because the majority of users will not change the default settings. Arthur Edelstein's philosophy is to have privacy by default for everyone and to have to accept trackers ourselves if we want to, not the other way around[13]. Also, it makes testing easier to do. However, some[14] retort that this puts some browsers at a disadvantage which, although they do not set privacy by default, provide options to choose from the start to protect their users' privacy if they choose to (with an ad blocker for example). These browsers will be considered as if they did not offer any privacy options, even though they can be easily enabled at startup. Moreover, before January 2022, the website's creator did not say on his website that the settings was the default ones, whether on the main page or even other accessible tabs like "About". This could mislead users who visit `PrivacyTests`. Nevertheless, after having made the remark, Arthur Edelstein agreed, and specified it at the top of the site. This shows once again his openness to the community and his involvement in this project.

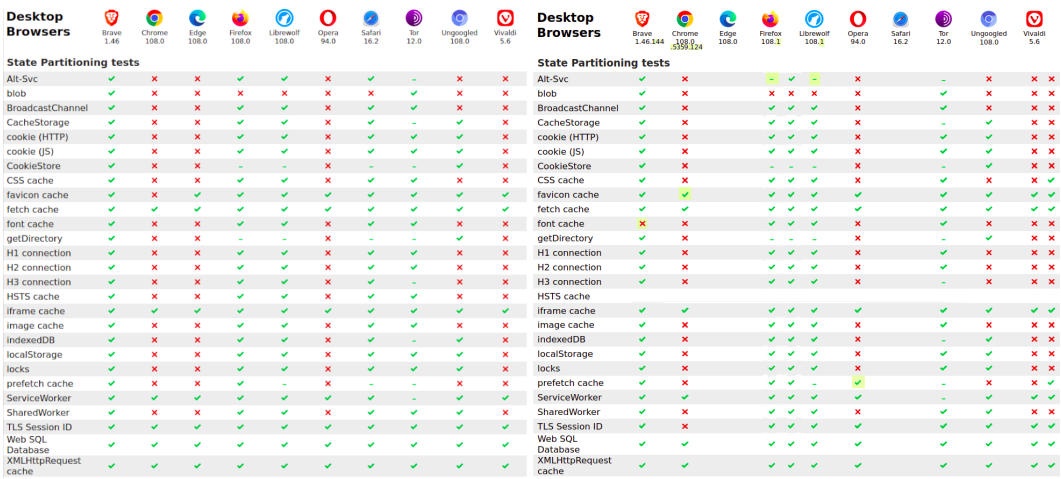
— At the time of the article, Arthur Edelstein works at Brave. a criticisms often given is that he could be inspired (voluntarily or not) by his work at Brave to make new tests, which will therefore be valid by default for this browser. One might believe that this is what happens in the section "Tracking query parameter tests" where he says himself in the description that "The set of tracking query parameters tested here was largely borrowed from Brave". But in reality, these tests and this sentence were already present in November 2021[15], while he only started working at Brave in May 2022. Since then, almost no category has changed. He is very transparent about his work and even displays it in the "about" section of the site.

Conversely, it is paid to "[...] work on fixing some of the privacy leaks that had been identified by the `privacytests.org` website." at Brave, like he said. This may bias the interpretation of the results a bit, as the tests are not necessarily exhaustive. But we can think that other browsers that have more passed tests over time may have used the same technique. This is a good thing overall for privacy if tests are relevant.

— The privacy tests do not take into account the additional services that Vivaldi has, even though they can improve the privacy of users. These tests are, for example, email or blog services without ads or tracking that are offered as an alternative to GAFAM, or feed aggregators that allow users to follow sites or YouTube channels without being tracked[16][17].

— Moreover, Vivaldi criticized the website [14] because they say they use another method to protect themselves from trackers which has the advantage of not breaking websites and which is not taken into account in the tests. This forces everyone to use the same method. The website author replied [18][12] that the method used by other browsers does not break sites and is very good, so there is no need to use anything else.

4. Replication of the Tests



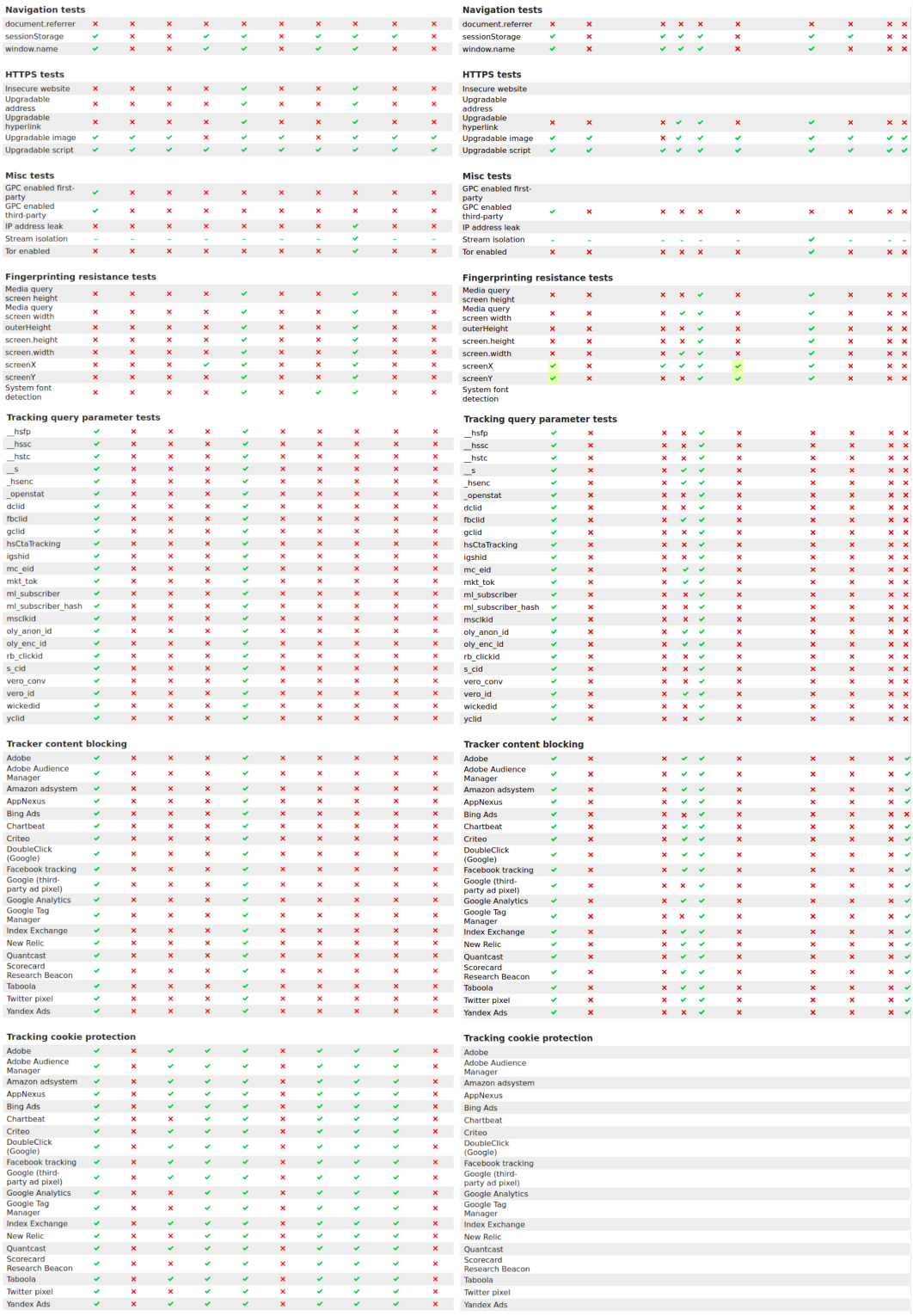


Figure 1: Original tests on the left and replicate tests on the right

In this section can be found the tests we managed to replicate on a macOS Catalina virtual machine. We were not able to run some of the tests or install some of the browsers, which may be due to the old version of macOS. The choice of macOS comes from the fact that it is the only OS to support all the browsers of the site, and in particular Safari. Note that since January 2023, it is possible to run the tests directly from Linux.

The differences between our tests and those of the site are highlighted in yellow. The differences are maybe due to the browser's version that is not exactly the same, and the version of macOS (Catalina) that may be different. But there are few differences, only 9 out of 680 tests.

We also did two tests with more advanced browser settings, on Firefox and Vivaldi. These two were the most interesting because we wanted to see if the numerous options of Firefox like "https only", "tracking strict", and "do not track" would give the same good results as LibreWolf which is a fork of Firefox (with privacy options, uBlock Origin, DuckDuckGo

and advanced options on fingerprints). We also wanted to see what would change if Vivaldi implemented as default settings the advanced privacy options like "Ask Websites Not to Track Me, Block Hyperlink Audit Tracking, Block Trackers and Ads, Never accept Cookies" (from the first connection, Vivaldi gives the choice to have an ad blocker and tracker blocker). We also tested other browsers without much success. Some browsers like Brave offer very few additional options, which did not change the results. And for Tor with the "safest" options, the tests were blocked. We notice that for Firefox the tests are much better with the options, but still far from LibreWolf, this can be due to the uBlock Origin plugin that LibreWolf has by default. Indeed, we can see in a tweet[19] that PrivacyTests had the same results as us with ETP strict turned on mode for the category "Tracker Content Blocking". It also shows the importance of the presence of plugins on the browser. This allows more freedom and choice in privacy, and they can significantly increase browser privacy, but in the other hand plugins can sell information themselves and therefore harm privacy. Finding and installing the right ones isn't easy for everyone, and that's why it seems like a good idea for LibreWolf to have uBlock Origin installed by default.

For Vivaldi, the privacy options are of little use, except the "Block Tracker Content" category (which was the default behaviour between February and June 2022). To improve other aspects of privacy, the user needs to switch to private browsing. There is an interesting observation when going back in time with the time machine[10]: Vivaldi passed a lot of tests at some point, then failed them, and passed them again in other places. Some of the features enabling to pass these tests became privacy options (such as "Tracker content blocking" that are not by default any more), some others are used in private browsing, and others disappear completely. This is very strange because it goes against what Vivaldi says. Indeed, they advocate the priority of privacy and say that they do not gain anything with people's data. One can say that some features "break" some websites, but this is not the case of the "Tracker content blocking" which does not change anything about the behaviour of the page, but just adds a tracker in the URL. So we have no clear explanation for this.

Also, we did not manage to test other browsers because the configuration of new tests is quite complex. We would have liked to try the list of browsers proposed in the `PrivacyTests` issue[20] (see section 6.2).

5. Conclusion

We have seen that `PrivacyTests.org` is an open-source website created by Arthur Edelstein, a passionate and invested professional in the field.
The website is rapidly gaining popularity since its creation at the end of 2021. It seems to be the only website to offer such a general browser comparison and in the most objective way possible with automatic tests. Its success makes it influential, for example the browsers improved by 70% on passing tests in 2022 on average. And although the website is criticized, which show the difficulty of making unbiased and exhaustive tests, the creator of the website manages to make a visual and accessible comparison to everyone on a very technical field. Obviously, this leads to simplifications that can make people think that the more green crosses there are, the better the browser is, which is not entirely true as we have seen. In addition to that, it remains difficult to interpret the results without knowledge of how browsers and tracking mechanisms work.
We were also able to replicate the tests and go a little further to see that the tests seem relevant.

References

[1] A. Edelstein, Privacytests.org (2022).
URL `https://privacytests.org`

[2] A. Edelstein, Arthur edelstein's linkedin experience.
URL `https://www.linkedin.com/in/arthuredelstein/details/experience`

[3] A. Edelstein, Arthur edelstein's youtube presentation: Tor browser and the fight for privacy (2017).
URL `https://youtu.be/Gl3nj8UfiGI`

[4] A. Edelstein, Privacytests.org's news (2022).
URL `https://https://privacytests.org/news.html`

[5] A. Edelstein, Privacytests.org's github (2022).
URL `https://github.com/arthuredelstein/privacytests.org`

[6] LibreWolf, Librewolf ask to not use the tor network.
URL https://librewolf.net/docs/faq/#can-i-use-librewolf-with-tor

[7] LibreWolf, Github issue: Make a list of desired fingerprinting tests (2018).
URL https://github.com/arthuredelstein/privacytests.org/issues/5

[8] Mozilla, Firefox blocks fingerprinting.
URL https://www.mozilla.org/en-US/firefox/features/block-fingerprinting

[9] Mozilla, Firefox's protection against fingerprinting.
URL https://support.mozilla.org/en-US/kb/firefox-protection-against-fingerprinting

[10] A. Edelstein, Wayback machine on privacytests.org (Nov. 2022).
URL https://web.archive.org/web/20230000000000*/http://privacytests.org/

[11] D. J. Leith, Web browser privacy: What do browsers say when they phone home? (2020).
URL https://www.scss.tcd.ie/Doug.Leith/pubs/browser_privacy.pdf

[12] A. Edelstein, Arthur edelstein responds to vivaldi on privacytests (Oct. 2022).
URL https://privacytests.org/vivaldi.html

[13] A. Edelstein, Arthur edelstein's explaining the need to clarify the definition for a consent to privacy (2022).
URL https://www.linkedin.com/feed/update/urn:li:ugcPost:6998393121290035200?commentUrn=urn%3Ali%3Acomment%3A%28ugcPost%3A6998393121290035200%2C6998401814706487296%29

[14] Vivaldi, Vivaldi reviews privacytests (Oct. 2022).
URL https://web.archive.org/web/20221003174703/https://vivaldi.com/security/common-questions/#privacytests

[15] A. Edelstein, Wayback machine november 2021 on privacytests.org (2021).
URL https://web.archive.org/web/20211106103619/https://privacytests.org

[16] Vivaldi, Blog service by vivaldi.
URL https://help.vivaldi.com/fr/services-fr/vivaldi-blogs-fr/creer-un-nouveau-blog-sur-vivaldi-net-et-ajouter-des-utilisateurs

[17] Vivaldi, Email service by vivaldi.
URL https://webmail.vivaldi.net

[18] A. Edelstein, Arthur edelstein responds to vivaldi on twitter (Oct. 2022).
URL https://twitter.com/privacytests/status/1578039734630117377

[19] A. Edelstein, Tests firefox with ublock origin or strict mode (Nov. 2022).
URL https://twitter.com/privacytests/status/1588567227133231105

[20] A. Edelstein, Privacytests.org's browsers propositions (2021).
URL https://github.com/arthuredelstein/privacytests.org/issues/66

6. Browsers

6.1. Tested Browsers

On the website, the browsers are divided between six categories:

1. Desktop browsers;
2. Desktop private modes;
3. iOS browsers;
4. Android browsers;
5. Nightly builds;
6. Nightly private modes.

In Table 1, we can see the browser version for each browser category.

Table 1: Browser versions

Browser	Desktop browsers	Desktop private modes	iOS browsers	Android browsers	Nightly builds	Nightly private modes
Brave	1.46	1.46 Private	1.45	1.46	1.48	1.48 Private
Bromite	—	—	—	108.0	—	—
Chrome	108.0	108.0 Private	108.5359	108.0	111.0	111.0 Private
Duckduckgo	—	—	7.70	5.145	0.30	—
Edge	108.0	108.0 Private	107.1418	108.0	110.0	110.0 Private
Firefox	108.0	108.0 Private	108.1	108.1	109.0a1	109.0a1 Private
Focus	—	—	108.0	108.1	—	—
Librewolf	108.0	108.0 Private	—	—	—	—
Mull	—	—	—	107.2	—	—
Opera	94.0	94.0 Private	3.4	72.5	96.0	96.0 Private
Safari	16.2	16.2 Private	16.1	—	16.4	16.4 Private
Samsung	—	—	—	19.0	—	—
Tor	12.0	12.0 Private	—	102.2	12.5a1	12.5a1 Private
Ungoogled	108.0	108.0 Private	—	—	—	—
Vivaldi	5.6	5.6 Private	—	5.6	5.7	5.7 Private
Yandex	—	—	2211.7	22.11	—	—

6.2. Additional Browsers to Test

In addition to the tested browsers, other browsers can be tested. In order to choose them, Edelstein created an issue on GitHub to ask the community for ideas of new browsers to test[20]. Some ideas given are browsers that are privacy-focused, which would also be interesting to test, at least to compare their results to the tests.

Here is the list of the proposed browsers in the issue:

- Arctic Fox: `https://github.com/rmottola/Arctic-Fox`,
- Avast: `https://www.avg.com/en/signal/best-browsers-most-security-privacy`,
- Decentr: `https://decentr.net/`,
- Dooble: `https://textbrowser.github.io/dooble/`,
- Dot: `https://www.dothq.co/en`,
- DuckDuckGo for Mac: `https://duckduckgo.com/mac`,
- Epic: `https://www.epicbrowser.com/`,
- Falkon: `https://www.falkon.org/`,
- Fennec: `https://f-droid.org/en/packages/org.mozilla.fennec_fdroid/`,
- Ghostery Dawn: `https://www.ghostery.com/private-browser`,
- GNOME web: `https://doc.ubuntu-fr.org/epiphany`,
- GNU IceCat: `https://www.gnu.org/software/gnuzilla/`,
- Hexavalent: `https://github.com/Hexavalent-Browser/Hexavalent`,
- Iceraven: `https://github.com/fork-maintainers/iceraven-browser`,
- Iodé: `https://iode.tech/en/`,
- Iridium: `https://iridiumbrowser.de/`,

- Iron: https://www.srware.net/iron/#downloads,
- Kagi (safari fork): https://kagi.com/,
- Kiwi Browser: https://kiwibrowser.com/,
- Librewolf: https://librewolf-community.gitlab.io/,
- Lynx: https://lynx.invisible-island.net/,
- Midori: https://astian.org/midori-browser/,
- Min: https://minbrowser.org/,
- Mulch: https://divestos.org/index.php?page=our_apps#mulch,
- Mull for Android: https://f-droid.org/en/packages/us.spotco.fennec_dos/,
- Onion: https://onionbrowser.com/,
- Orion browser: https://browser.kagi.com/,
- Otter: https://otter-browser.org/,
- Pale Moon: https://palemoon.org,
- Privacy Browser: https://f-droid.org/en/packages/com.stoutner.privacybrowser.standard/,
- SeaMonkey: https://www.seamonkey-project.org/,
- Sidekick: https://www.meetsidekick.com/,
- SnowHaze on iOS: https://snowhaze.com/fr/index.html,
- SpiderWeb: https://github.com/wicknix/SpiderWeb,
- Surf Browser: https://surf.suckless.org/,
- Ungoogled chromium & Ungoogled Chromium Android : https://ungoogled-software.github.io/ & https://github.com/ungoogled-software/ungoogled-chromium-android/releases,
- UR-browser: https://www.ur-browser.com/,
- Vanadium: https://github.com/GrapheneOS/Vanadium,
- vivaldi: https://vivaldi.com/,
- Waterfox: https://www.waterfox.net/,
- Yandex desktop: https://browser.yandex.com/beta.

From this list, only Librewolf was tested in the project. According to the creator of the site, this is because it had a unique combination of privacy protections, which might be of interest to users, and shows what is possible for a web browser to achieve"

7. Tested items

7.1. Item Categories

There are eight item categories in total:

- State Partitioning tests
- Navigation tests
- HTTPS tests
- Misc tests
- Fingerprinting resistance tests
- Tracking query parameter tests

- Tracker content blocking
- Tracking cookie protection

Depending on the browser category, the items categories may vary. In Table 11, we can see which item categories are tested for each browser category.

Table 2: Item Categories Depending on Browser Category

Categories	Desktop browsers	Desktop private modes	iOS browser	Android browsers	Nightly builds	Nightly private modes
State Partitioning tests	x	x	x	x	x	x
Navigation tests	x	x	x	x	x	x
HTTPS tests	x	x	x	x	x	x
Misc tests	x	x	x	x	x	x
Fingerprinting resistance tests	x	x	x	x	x	x
Tracking query parameter tests	x	x	x	x	x	x
Tracker content blocking	x	x	x	x	x	x
Tracking cookie protection	x	x	—	—	x	x

In most cases, the items present in one category are always the same. There are only two exceptions

- The iOS and Android browsers categories do not have the Tracking cookie protection category.
- In the fingerprinting resistance test category, the item System font detection isn't tested for iOS and Android browsers.

7.2. Items Detailed

7.2.1. State Partitioning tests

Often, web browsers allow tracking companies to 'tag' the browser with some data ('state') that identifies you and that can be seen by third-party trackers embedded in websites. This can be solved by partitioning all data stored in the browser. In other words, we restrict the sharing of information between websites, which makes cross-site tracking more complicated.

We can take cookie partitioning as an example:

- Without cookie partitioning, if a website A creates a cookie for the website C, and a website B wants to access the cookie for the website C, it will work without a problem. Indeed, there is a cookie for the website C that can be accessed by any website.
- With state partitioning, a unique identifier will be created for the cookie for the website C accessed by the website A and another for the cookie for the website C accessed by the website B. Therefore, the cookie information won't be shared between the websites A and B, even if the cookies are for the same website C.

This category is called state partitioning and not cookie partitioning because it also includes this mechanism for anything that is shared: cookies, local storage, session storage, cache storage and indexed databases. Some of the subcategories of State Partitioning are:

- Networking State Partitioning
- Cookies State Partitioning
- Storage State Partitioning
- Assorted State Partitioning

Table 3: State Partitioning Tests (1)

Item	Description
Alt-Svc	The Alt-Svc (Alternative Services) header is used to indicate to the server that a resource should be loaded on a different server or network location, which is the alternative service. However, it's a persistent setting and can therefore also be used to track users across websites if the partition is not correctly done.
blob	A blob URL (Binary Large Object URL) can be used to share the raw data referenced by the blob between websites by trackers. If they are not revoked, they cannot be garbage collected. It is therefore important to partition, so that the websites cannot share information with blobs.
BroadcastChannel	The BroadcastChannel interface represents a named channel that is used to communicate between different documents of the same origin. In other words, it allow windows, tabs, frames to communicate with each other. If it is not correctly partitioned, it can be used for cross-site communication, but also for tracking.
CacheStorage	The CacheStorage interface refers to the storage of Cache objects. It has a main directory with all the named caches that can be accessed by the ServiceWorkers or other workers. If it is not correctly partitioned, the same Cache object can be accessed to multiple websites, which can be used to track users.
cookie (HTTP)	When a client sends a request for the first time, the server can attach one or more cookies with its response. With future requests, the client can attach the cookies to retrieve the same state between cookies. These cookies are stored locally on the client's side. This can easily be used to track the user, and partitioning it limits this tracking by only allowing the site corresponding to the cookie to retrieve it. In other words, a site cannot retrieve a cookie that was sent by another website. An HTTP-only cookie cannot be accessed in JavaScript.
cookie (JS)	Similarly to the HTTP cookie, a JS cookie is stored on the computer to keep information about the user and send it back when needed. However, its difference with HTTP-only cookies is that it can be accessed (read from and written to) in client side JavaSscript as well as in server side.
CookieStore	The Cookie Store API is used to get and set cookies asynchronously from a page or a service worker. By partitionning the storage for CookieStore, we can prevent sites from accessing the data of the cookies set by other sites.
CSS cache	To prevent having to load the CSS stylesheet multiple times, the browser can cache the stylesheet. If this cache is shared between websites, it can be used for tracking across sites.
favicon cache	The favicon is the icon used to represent a website. We can partition the favicon cache so that a website can't have access to all the favicons stored in the browser's cache.
fetch cache	When a resource is fetched with the Fetch API, it can be cached so that the browser doesn't have to fetch it multiple times. We can partition this cache to prevent websites to access the cache for other websites and do cross-site tracking.
font cache	The browser can store cache on the web fonts, which can be used for cross-site tracking.

Table 4: State Partitioning Tests (2)

Item	Description
getDirectory	The method navigator.storage.getDirectory can be used to expose a location for storing files to web content. In some cases, these files may be shared across tabs. This allows to locate the files containing user data from other websites if there is no partitioning.
H1 connection	HTTP/1.x are the classic web connection protocols. If these connections are re-used across websites, they can be used to track users.
H2 connection	HTTP/2 is a web connection protocol introduced in 2015. Some browsers re-use HTTP/2 connections across websites and can thus be used to track users.
H3 connection	HTTP/3 is a new standard HTTP connection protocol, still in draft but widely supported by browsers. If it is not partitioned, it can be used to track users across websites.
HSTS cache	HTST (HTTP Strict-Transport-Security) is a standard that ensures the browser always connects to a website over HTTPS. It enables websites to signal that they should only be accessed via HTTPS, which is stored in a database. If this database is not partitioned, then it can be used to track users across websites.
iframe cache	The iframe (Inline Frame) element in a web page allows websites to embed an HTML page in the current one. Caching of this web page could be used for cross-site tracking.
image cache	Browsers often cache images to reload websites faster. If the cached images are accessed by other websites, it can be used for cross-site tracking.
indexedDB	The indexedDB API is used for client-side storage of data, which can be used for cross-site tracking.
localStorage	The localStorage API allows websites to store data accross browser sessions, which can be used for cross-site tracking if it's not properly partitioned.
locks	The Web Locks API allows tabs to communicate for script coordination, which can be used for cross-site tracking if it is not partitioned.
prefetch cache	Cache prefetching is used by browsers to load resources faster by fetching them in advance and caching them.
ServiceWorker	The ServiceWorker API acts essentially as a proxy server between web applications, the browser and the network to store the content so that it can be accessed even when the user is offline. If it can be accessed by multiple browsers, it can be used for cross-site tracking.
SharedWorker	The SharedWorker API is used to share data across different browsing context such as windows.
TLS Session ID	If the browser re-uses a TLS session, the session ID can be used for cross-site tracking.
Web SQL Database	The Web SQL Database API is deprecated and should not be used.
XMLHttpRequest cache	XMLHttpRequest objects are used to update part of a website without disrupting the user activity. If the cached resources are shared between sites, it can be used for tracking.

7.2.2. Navigation tests

When the user accesses a website by clicking on a link from another website, some browser APIs allow the first site to communicate to the second one. This is used for exchanging useful data and in some functionalities, but it is also very useful for cross-site tracking. This privacy vulnerability can be fixed by introducing new limits on how much data is transfered

between sites.

Table 5: Navigation tests

Item	Description
document.referrer	The Referrer request header is a mechanism used by browsers to let a website know where the user is visiting from. This header is inherently tracking users across websites. In recent times, browsers have switched to a policy of trimming a referrer to convey less tracking information, but Referer continues to convey cross-site tracking data by default. This data can be reduced but never to 0% because else, some websites would not work correctly. However some browser allow the user to to minimize it with options (Opera, Comodo Dragon and Firefox).
sessionStorage	The sessionStorage API is similar to the localStorage API, but it does not persist across tabs or across browser sessions. Nonetheless, it can be used to track users if they navigate from one website to another. This tracking can be thwarted by partitioning sessionStorage between websites.
window.name	The window.name API allows websites to store data that will persist after the user has navigated the tab to a different website. This mechanism could be partitioned so that data is not allowed to persist between websites.

7.2.3. HTTPS tests

The items in the category HTTPS tests allow to test if HTTPS is being used. Indeed, with HTTPS the connection is encrypted so that third-parties cannot read the communications between the server and the browser, as opposed to insecure connections that were used by default in the past.

Table 6: HTTPS tests

Item	Description
Insecure website	Tests if websites that do not use HTTPS are loaded or if a warning is given to the user. The alternative is the cross on the lock in the URL, which is less visible and does not stop the insecure connection.
Upgradable address	Tests if the browser automatically changes the address from HTTP to HTTPS when it is possible.
Upgradable hyperlink	The same for hyperlinks.
Upgradable image	The same for images.
Upgradable script	The same for scripts.

7.2.4. Misc tests

This category tests for the presence of miscellaneous privacy features.

Table 7: Misc tests

Item	Description
GPC enabled first-party	The GPC (Global Privacy Control) is an HTTP header which sends the instruction not to sell users' personal data to third parties. This tests if the browser sends this header by default to the websites, however this does not ensure that the websites respect this instruction.
GPC enabled third-party	The same as GPC enabled first-party but for sending the header to third-party elements on the web page.
IP address leak	This tests if the browser conceals the user's IP address from the websites with a proxy, a VPN or the Tor network.
Stream isolation	With stream isolation, each connection uses a new circuit. That can be done using a proxy for example. With Tor, browsers can use a different Tor circuit per website.
Tor enabled	Tests if the Tor network is being used by default. The Tor network sends the requests through a series of relays to hide the user's IP address. This enhances anonymity online, but is specific to the Tor network.

7.2.5. Fingerprinting resistance tests

Fingerprinting is a technique that identifies users by using scripts measuring their environment's settings and characteristics. The set of measurements will build a signature for the user.

Table 8: Fingerprinting resistance tests

Item	Description
Media query screen height	Media queries are used to apply CSS styles depending on the device and browser characteristics. That way, the websites can learn the device's screen height.
Media query screen width	The same as the previous item but with the width.
outerHeight	The jQuery method outerHeight() returns the outer height of the first matched element in the page. This can give information on the screen's height to websites.
screen.height	The CSS Object Model allows users to read and modify CSS style dynamically. The screen.height property in this API returns the height of the screen in pixel, which can be used for fingerprinting.
screen.width	The same as previously but for the width.
screenX	The Window.screenX property returns the horizontal (x) coordinate of a window relative to the screen in CSS pixels.
screenY	The same as previously but vertically (y).
System font detection	Web pages can detect which fonts are installed on the user's system. This can be used for fingerprinting because different browsers, browser versions, or systems, support different sets of fonts.

7.2.6. Tracking query parameter tests

To track users, websites can pass values in the URL by adding query string parameters to them. We call these **tracking query parameters** and they can be synchronized with cookies and contain unique identifier for the users, which make them into very powerful tracking tools. However, web browsers can delete them from the URLs before they are sent, which is tested in this category.

Table 9: Tracking query parameter tests (1)

Item	Description
___hsfp	HubSpot Fingerprint, no further information was found on it. HubSpot is a marketing and sales software.
___hssc	HubSpot Session Cookie, used to keep track of sessions by storing the domain, the view count and the session start timestamp.
___hstc	HubSpot Tracking Cookie, the main cookie for tracking visitor. Contains the domain, the utk, the initial, last and current timestamps, and the session number.
___s	Drip.com email address tracking parameter, used to track and identify the people who click a link in an email.
_hsenc	HubSpot Enccryption, no further information was found on it.
_openstat	Yandex tracking parameter
dclid	DoubleClick Click ID (Google), used to track which ads the user clicks.
fbclid	Facebook Click Identifier.
gclid	Google Click Identifier, used to exchange data between Google Analytics and Google AdSense.
hsCtaTracking	HubSpot CTA (Calls-to-Action) Tracking parameter, specific to CTA tracking. In web design, a CTA is an element that a user clicks to continue to the next step when buying an article.
igshid	Instagram Share Id, used by Instagram to track shares from a profile.
mc_eid	Mailchimp Email ID, used by Mailchimp, a marketing platform, to track clicks which come from email marketing campaigns. It contains the user id which Mailchimp passes for tracking the user-level conversion metrics.
mkt_tok	Adobe Marketo tracking parameter, vital for tracking web session activities by identifying when the user has clicked a Marketo email link.
ml_subscriber	MailerLite email subscriber tracking, tracks email campaign activity and subscriber engagement.
ml_subscriber_hash	MailerLite email subscriber hash tracking, similar as the previous item.
msclkid	Microsoft Click Identifier is a click identification that is automatically added by Microsoft Advertising to link URLs.
oly_anon_id	Olytics (Omeda's web analytics service) Anonymous Record Id, created when a new user visits an Olytics site.
oly_enc_id	Olytics Encrypted Customer Id, created if the user is known with Olytics.
rb_clickid	Unknown high-entropy tracking parameter, found in many Russian websites.
s_cid	Adobe Site Catalyst tracking parameter, used for Adobe Analytics.
vero_conv	Vero Conversion tracking parameter, used to track the conversions from an email campaign.
vero_id	Vero Id tracking parameter, used to track who clicks on links by passing their user ID.
wickedid	Wicked Reports e-commerce tracking, used to track clicks associated with specific Facebook ads.
yclid	Yandex Click ID, communicates the unique click number of a Yandex ad.

7.2.7. Tracker content blocking

Some websites can track the user by using third-party embedded components such as scripts and tracking pixels. This category tests if browsers block the websites' tracking components.

The items tested are said to correspond to the twenty largest trackers listed by `https://whotracks.me`: Google Static, Google Tag Manager, Google Analytics, Google, DoubleClick (Google), Google Fonts, Google APIs, Facebook, YouTube, Amazon CloudFront, Amazon Advertising, Google User Content, Google Syndication, Google Photos, CloudFlare, ScoreCard Research Beacon (comScore, Inc.), Amazon Web Services, Twitter, jsDelivr and Amazon CDN Amazon.

However, the lists are not the same. The list was probably taken from a few years ago.

Each item tests if the URL corresponding to the tracker is blocked from loading. In Table 10 are listed the items and corresponding URLs that should be blocked.

Table 10: Tracker content blocking tests

Item	Description
Adobe	`https://munchkin.marketo.net/munchkin.js`
Adobe Audience Manager	`https://dpm.demdex.net/ibs`
Amazon adsystem	`https://s.amazon-adsystem.com/dcm`
AppNexus	`https://ib.adnxs.com/px?id=178248&t=1`
Bing Ads	`https://bat.bing.com/bat.js`
Chartbeat	`https://static.chartbeat.com/js/chartbeat.js`
Criteo	`https://dis.criteo.com/dis/rtb/appnexus/cookiematch.aspx`
DoubleClick (Google)	`https://securepubads.g.doubleclick.net/static/glade.js`
Facebook tracking	`https://connect.facebook.net/en_US/fbevents.js`
Google (third-party ad pixel)	`https://www.google.com/pagead/1p-user-list/`
Google Analytics	`https://google-analytics.com/urchin.js`
Google Tag Manager	`https://www.googletagmanager.com/gtag.js?id=GTM-NX4SMZL`
Index Exchange	`https://dsum-sec.casalemedia.com/crum?cm_dsp_id=10&external_user_id=629685505537&C=1`
New Relic	`https://js-agent.newrelic.com/nr-1212.min.js`
Quantcast	`https://pixel.quantserve.com/pixel`
Scorecard Research Beacon	`https://sb.scorecardresearch.com/internal-c2/default/cs.js`
Taboola	`https://trc.taboola.com/futureplc-tomsguide/trc/3/json`
Twitter pixel	`https://t.co/i/adsct`
Yandex Ads	`https://yandex.ru/ads/system/header-bidding.js`

7.2.8. Tracking cookie protection

Web pages can use hidden third-party trackers to read and write cookies that track the user's browsing across websites.

The items names in this category are the same as in the previous one. In this category, the items test whether the browser stops cookies from the corresponding link from tracking users across websites.

Table 11: Fingerprinting resistance tests

Item	Description
Adobe	`munchkin.marketo.net`
Adobe Audience Manager	`dpm.demdex.net`
Amazon adsystem	`s.amazon-adsystem.com`
AppNexus	`ib.adnxs.com`
Bing Ads	`bat.bing.com`
Chartbeat	`static.chartbeat.com`
Criteo	`dis.criteo.com`
DoubleClick (Google)	`securepubads.g.doubleclick.net`
Facebook tracking	`connect.facebook.net`
Google (third-party ad pixel)	`www.google.com`
Google Analytics	`google-analytics.com`
Google Tag Manager	`www.googletagmanager.com`
Index Exchange	`dsum-sec.casalemedia.com`
New Relic	`js-agent.newrelic.com`
Quantcast	`pixel.quantserve.com`
Scorecard Research Beacon	`sb.scorecardresearch.com`
Taboola	`trc.taboola.com`
Twitter pixel	`t.co`
Yandex Ads	`yandex.ru`

8. Evolution of Test Results

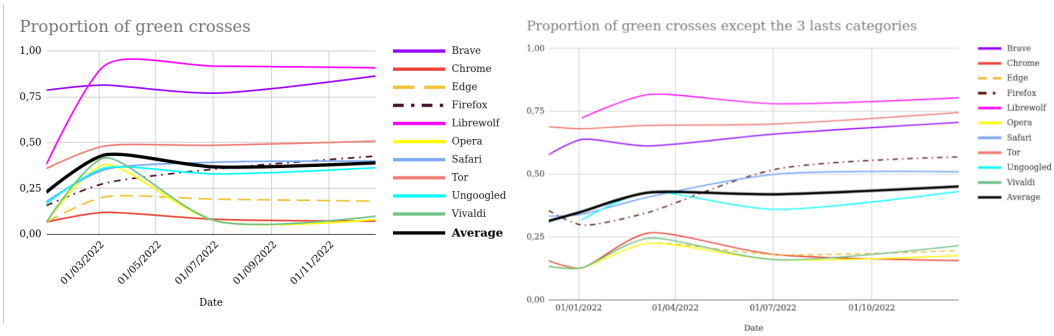


Figure 2: The proportion of green crosses in 2022

On the left, we can see the evolution of the proportion of tests passed on browsers and on average. On average there is a 70% increase of the passed tests. It should be noted that the number of tests has increased from 89 to 110 in 2022, which makes the evolution more difficult to measure. The statistics do not start before because the first date corresponds to the arrival of the Ungoogled and LibreWolf browsers, which would distort the average if we had added lower dates. We also added the image on the right with the same data, without the last 3 categories which have more than half of the tests, and therefore hide the evolution of the first ones. In this case, we go from 47 to 51 tests with a gain of passed tests of 27%.

It should be kept in mind, however, that, as the author said: "Unfortunately, if a browser fails to protect against a small number of privacy leaks (or often a single one), it is possible

for a user to be individually tracked across the web. So I tend to think of the red Xs as warning lights."

9. Acknowledgement

We would like to thank our professor, Cedric Lauradou, for allowing us to write this article.

We would like to express our gratitude to the author of the website privacytests.org, Arthur Edelstein, for answering all our questions.