Browser Fingerprinting: An Introduction and the Challenges Ahead

by gk (/users/gk) | September 04, 2019

Guest post by Pierre Laperdrix (https://plaperdr.github.io/)

In the past few years, a technique called browser fingerprinting has received a lot of attention because of the risks it can pose to privacy. What is it? How is it used? What is Tor Browser doing against it? In this blog post, I'm here to answer these questions. Let's get started!

What Is Browser Fingerprinting?

Since the very beginning of the web, browsers did not behave the exact same way when presented with the same webpage: some elements could be rendered improperly, they could be positioned at the wrong location or the overall page could simply be broken with an incorrect HTML tag. To remedy this problem, browsers started including the "user agent" header. This informed the server on the browser being used so that it could send the device a page that was optimized for it. In the nineties, this started the infamous era of the "Best on IE" or "Optimized for Netscape."

In 2019, the user-agent header is still here but a lot has changed since then. The web as a platform is a lot richer in terms of features. We can listen to music, watch videos, have real-time communications or immerse ourselves in virtual reality. We can also use a very wide variety of devices from tablets, smartphones or laptops to connect to it. **To offer an experience that is optimized for every device and usage, there is still a need today to share configuration information with the server.** "Here is my timezone so that I can know the exact start time of the NBA finals. Here is my platform so that the website can give me the right version of the software I'm interested in. Here is the model of my graphic card so that the game I'm playing in my browser can chose graphic settings for me."

All of this makes the web a truly beautiful platform as it enables us to have a comfortable experience browsing it. However, all that information that is freely available to optimize the user experience can be collected **to build a browser fingerprint**.

Figure 1: Example of a browser fingerprint from a Linux laptop running Firefox 67

In Figure 1, you can see a browser fingerprint taken from my Linux laptop. The information in the fingerprint was collected **via HTTP with the received HTTP headers and via JavaScript by running a small script.** The “user-agent” indicates that the user was using Firefox version 67 on the Fedora Linux distribution. The “content-language” header indicates that the user wants to receive her page in English with the “US” variant. The “-120” for the timezone refers to the GMT+2 time. Finally, the WebGL renderer gives information on the CPU of the device. Here, the laptop is using an Intel CPU with a Kaby Lake Refresh microarchitecture.

This example is a glimpse of what can be collected in a fingerprint and the exact list is evolving over time as new APIs are introduced and others are modified. If you want to see your own browser fingerprint, I invite you to visit AmIUnique.org (https://amiunique.org). It is a website that I launched in 2014 to study browser fingerprinting. With the data that we collected from more than a million visitors, we got invaluable insight into its inner-workings and we pushed the research in the domain forward.

What Makes Fingerprinting A Threat To Online Privacy?

It is pretty simple. First, there is **no need to ask for permissions** to collect all this information. Any script running in your browser can silently build a fingerprint of your device without you even knowing about it. Second, **if one attribute of your browser fingerprint is unique or if the combination of several attributes is unique, your device can be identified and tracked online**. In that case, no need for a cookie with an ID in it, the fingerprint is enough. Hopefully, as we will see in the next sections, a lot of progress have been made to prevent users from having unique values in their fingerprint and thus, avoid tracking.

Tor + Fingerprinting

Tor Browser was the very first browser to address the problems posed by fingerprinting as soon as 2007, even before the term “browser fingerprinting” was coined. In March 2007, the changelog for the Tor button indicated the inclusion of Javascript hooking to mask timezone for Date Object (https://gitweb.torproject.org/torbutton.git/tree/src/CHANGELOG).

In the end, the approach chosen by Tor developers is simple: **all Tor users should have the exact same fingerprint**. No matter what device or operating system you are using, your browser fingerprint should be the same as any device running Tor Browser (more details can be found in the Tor design document (https://2019.www.torproject.org/projects/torbrowser/design/#fingerprinting-linkability)).

Figure 2: Example of a browser fingerprint from a Linux laptop running Tor Browser 8.5.3

In Figure 2, you can find the fingerprint of my Linux machine running version 8.5.3 of the Tor Browser.

Comparing with the one from Firefox, we can see notable differences. First, no mater on which OS Tor Browser is running, you will always have the following user-agent:

```
Mozilla/5.0 (Windows NT 6.1; rv:60.0) Gecko/20100101 Firefox/60.0
```

As Windows is the most widespread OS on the planet, TBB masks the underlying OS by claiming it is running on a Windows machine. Firefox 60 refers to the ESR version on which TBB is based on.

Other visible changes include the platform, the timezone, and the screen resolution.

Also, you may have wondered why the following message appears when you maximize the browser window (see Figure 3): “Maximizing Tor Browser can allow websites to determine your monitor size, which can be used to track you. We recommend that you leave Tor Browser windows in their original default size.”

This is because of fingerprinting. Since users have different screen sizes, one way of making sure that no differences are observable is to have everyone use the same window size. If you maximize the browser window, you may end up as being the only one using Tor Browser at this specific resolution and so comes a higher identification risk online.

Figure 3: Warning from the Tor Browser when maximizing the browser window

Under the hood, a lot more modifications have been performed to reduce differences between users. Default fallback fonts (https://trac.torproject.org/projects/tor/ticket/18097) have been introduced to mitigate font and canvas fingerprinting. WebGL and the Canvas API are blocked by default to prevent stealthy collection of renderings. Functions like performance.now (https://trac.torproject.org/projects/tor/ticket/1517) have also been modified to prevent timing operations in the browser that can be used for micro-architectural attacks. If you want to see all the efforts made by the Tor team behind the scenes, you can take a look at the fingerprinting (https://trac.torproject.org/projects/tor/query?keywords=~tbb-fingerprinting&order=priority) tag in the bug tracker. A lot of work is being done to make this a reality. As part of the effort to reduce fingerprinting, I also developed a fingerprinting website called FP Central (https://fpcentral.tbb.torproject.org/) to help Tor developers find fingerprint regressions between different Tor builds.

Finally, more and more modifications present in TBB are making their way into Firefox as part of the Tor Uplift program (https://wiki.mozilla.org/Security/Tor_Uplift).

Where We Are

Over the past few years, research on browser fingerprinting has substantially increased and covers many aspects of the domain. Here, we will have a quick overview of the research done in academia and how fingerprinting is used in the industry.

Academic Research

1. Tracking with fingerprinting is a reality but it cannot replace different tracking schemes based on identifiers. Different studies have been published over the years trying to assess the diversity of modern devices connected on the web [1,2]. One study that I was part of in 2018 [3] surprised us as it showed that tracking at a very large scale may not be feasible with low percentage of uniqueness. Anyhow, the one clear takeaway from these studies is the following: even though some browser vendors are working very hard to reduce as much as possible the differences between devices, it is not a perfect process. If you have that one value in your browser fingerprint (or a combination) that nobody has, you can still be tracked and that is why you should be careful about fingerprinting. There is no strong guarantee today that your device is identical to another one present on the Internet.

2. As the web is getting richer, new APIs make their way into browsers and new fingerprinting techniques are discovered. The most recent techniques include WebGL [4,5], Web Audio [6] and extension fingerprinting [7,8]. To provide protection for users, it is important to keep a close watch on any new advances in the field to fix any issues that may arise.

One lesson learned from the past concerns the BatteryStatus API. It was added to provide information about the state of the battery to developers so that they could develop energy-efficient applications. Drafted as early as 2011, it was not until 2015 that researchers discovered that this API could be misused to create a short-term identifier [9,10]. In the end, this was a reminder that we have to be very thoughtful when introducing a new API in a browser. A deep analysis must be conducted to remove or mitigate as much as possible hidden fingerprinting vectors before they are deployed to end-users. To provide guidance for Web specification authors, the W3C has written a document (https://www.w3.org/TR/fingerprinting-guidance/) on how best to design an API while considering fingerprinting risks

Figure 4: Example of a WebGL rendering as tested on http://uniquemachine.org/ (http://uniquemachine.org/)

Figure 5: Example of an audio fingerprint as tested on https://audiofingerprint.openwpm.com/ (https://audiofingerprint.openwpm.com/)

3. Today, there is no ultimate solution to fix browser fingerprinting. As its origin is rooted in the beginning of the internet, there is no single patch that can fix it for good. And as such, designing defenses is hard. A lot of approaches have been tried and evaluated over the years with each their strengths and weakness. Examples include blocking attributes, introducing noise, modifying values, or increasing fingerprint diversity. However, one important observation that has been made is that sometimes having no specific defense is better than having one. Some solutions, because of the way they were designed or coded, remove some fingerprinting vectors but introduce some artifacts or inconsistencies in the collected fingerprints.

For example, imagine a browser extension that changes the value of fingerprints before they are sent.

Everything works perfectly except the fact that the developer forgot to override the navigator.platform value. Because of this, the user-agent may say that the browser is running on Windows whereas the platform still indicates it is on a Linux system. This creates a fingerprint that is not supposed to exist in reality and, as such, make the user more visible online. It is what Eckersley [1] called the “Paradox of Fingerprintable Privacy Enhancing Technologies.” By wanting to increase online privacy, you install extensions that in the end make you even more visible than before.

Industry

1. To identify websites who use browser fingerprinting, one can simply turn to privacy policies. Most of the time, you will never see the term “fingerprinting” in it but sentences along the lines of “we collect device-specific information to improve our services.” The exact list of collected attributes is often imprecise and the exact use of that information can be very opaque ranging from analytics to security to marketing or advertising.

Another way of identifying websites using fingerprinting is to look directly at the scripts that run in the browser. The problem here is that it can be challenging to differentiate a benign script that is here to improve the user experience from a fingerprinting one. For example, if a site accesses your screen resolution, is it to adjust the size of HTML elements to your screen or is it the first step in building a fingerprint of your device? The line between the two can be very thin and identifying fingerprinting scripts with precision is still a subject that has not been properly studied yet.

2. One use of fingerprinting that is lesser known is for bot detection. To secure their websites, some companies rely on online services to assess the risk associated with external connections. In the past, most decisions to block or accept a connection was purely based on IP reputation. Now, browser fingerprinting is used to go further to detect tampering or identify signs of automation. Examples of companies that use fingerprinting for this purpose include ThreatMetrix, Distil Networks, MaxMind, PerimeterX, and DataDome.

3. On the defensive side, more and more browser vendors are adding fingerprinting protection directly in their browser. As mentioned previously in this blog post, Tor and Firefox are at the forefront of these efforts by limiting passive fingerprinting and blocking active fingerprinting vectors.

Since its initial release, the Brave browser also includes built-in protection (https://github.com/brave/browser-laptop/wiki/Fingerprinting-Protection-Mode) against it.

Apple made changes to Safari in 2018 to limit it (https://gizmodo.com/apple-declares-war-on-browser-fingerprinting-the-sneak-1826549108) and Google announced in May 2019 its intention to do the same for Chrome (https://blog.chromium.org/2019/05/improving-privacy-and-security-on-web.html).

Conclusion: What Lies Ahead

Browser fingerprinting has grown a lot over the past few years. As this technique is closely tied to browser technology, its evolution is hard to predict but its usage is currently shifting. What we once thought could replace cookies as the ultimate tracking technique is simply not true. Recent studies show that, while it can be used to identify some devices, it cannot track the mass of users browsing the web daily. Instead, fingerprinting is now being used to improve security. More and more companies find value in it to go beyond traditional IP analysis. They analyze the content of fingerprints to identify bots or attackers and block unwanted access to online systems and accounts.

One big challenge surrounding fingerprinting that is yet to be solved is around the regulation of its usage. For cookies, it is simple to check if a cookie was set by a specific website. Anyone can go in the browser preferences and check the cookie storage. For fingerprinting, it is a different story. There is no straightforward way to detect fingerprinting attempts and there are no mechanisms in a browser to completely block its usage. From a legal perspective, this is very problematic as regulators will need to find new ways to cooperate with companies to make sure that the privacy of users is respected.

Finally, to finish this post, is fingerprinting here to stay? In the near future at least, yes. This technique is so rooted in mechanisms that exist since the beginning of the web that it is very complex to get rid of it. It is one thing to remove differences between users as much as possible. It is a completely different one to remove device-specific information altogether. Only time will tell how fingerprinting will change in the coming years but its evolution is something to watch closely as the frantic pace of web development will surely bring a lot of surprises along the way.

Thanks a lot for reading this post all the way through! If you want to dive even deeper in the subject, I invite you to read the survey (https://arxiv.org/pdf/1905.01051) [11] on the topic that we recently made available online. If by any chance you find any new fingerprinting vectors in Tor Browser, I strongly suggest that you open a ticket on the Tor bug tracker (https://trac.torproject.org/projects/tor) to help the fantastic efforts made by the Tor dev team to better protect users' online anonymity!

Pierre Laperdrix
https://plaperdr.github.io/ (https://plaperdr.github.io/)
Twitter: https://twitter.com/RockPartridge (https://twitter.com/RockPartridge)

References

[1] P. Eckersley. “How unique is your web browser?”. In International Symposium on Privacy Enhancing Technologies Symposium (PETS’10). [[PDF]](https://panopticlick.eff.org/static/browser-uniqueness.pdf (https://panopticlick.eff.org/static/browser-uniqueness.pdf))

[2] P. Laperdrix, W. Rudametkin and B. Baudry. “Beauty and the Beast: Diverting Modern Web Browsers to Build Unique Browser Fingerprints”. In IEEE Symposium on Security and Privacy (S&P’16). [[PDF]](https://hal.inria.fr/hal-01285470v2/document (https://hal.inria.fr/hal-01285470v2/document))

[3] A. Gómez-Boix, P. Laperdrix, and B. Baudry. “Hiding in the Crowd: an Analysis of the Effectiveness of Browser Fingerprinting at Large Scale”. In The Web Conference 2018 (WWW’18). [PDF (https://hal.inria.fr/hal-01718234v2/document)]

[4] K. Mowery, and H. Shacham. “Pixel perfect: Fingerprinting canvas in HTML5”. In Web 2.0 Security & Privacy (W2SP’12). [PDF (https://www.ieee-security.org/TC/W2SP/2012/papers/w2sp12-final4.pdf)]

[5] Y. Cao, S. Li, and E. Wijmans. “(Cross-) Browser Fingerprinting via OS and Hardware Level Features”. In Network and Distributed System Security Symposium (NDSS’17). [PDF (https://www.ndss-symposium.org/wp-content/uploads/2017/09/ndss2017_02B-3_Cao_paper.pdf)]

[6] S. Englehardt, and A. Narayanan. “Online tracking: A 1-million-site measurement and analysis”. In ACM SIGSAC Conference on Computer and Communications Security (CCS’16). [PDF (http://randomwalker.info/publications/OpenWPM_1_million_site_tracking_measurement.pdf)]

[7] A. Sjösten, S. Van Acker, and A. Sabelfeld. “Discovering Browser Extensions via Web Accessible Resources”. In ACM on Conference on Data and Application Security and Privacy (CODASPY’17). [PDF (https://www.cse.chalmers.se/~andrei/codaspy17.pdf)]

[8] O. Starov, and N. Nikiforakis. “XHOUND: Quantifying the Fingerprintability of Browser Extensions”. In IEEE Symposium on Security and Privacy (S&P’17). [PDF (https://securitee.org/files/xhound-oakland17.pdf)]

[9] Ł. Olejnik, G. Acar, C. Castelluccia, and C. Diaz. “The Leaking Battery”. In International Workshop on Data Privacy Management (DPM’15). [PDF (https://eprint.iacr.org/2015/616.pdf)]

[10] Ł. Olejnik, S. Englehardt, and A. Narayanan. “Battery Status Not Included: Assessing Privacy in Web Standards”. In International Workshop on Privacy Engineering (IWPE’17). [PDF (http://randomwalker.info/publications/battery-status-case-study.pdf)]

[11] P. Laperdrix, N. Bielova, B. Baudry, and G. Avoine. “Browser Fingerprinting: A survey”. [PDF - Preprint (https://arxiv.org/pdf/1905.01051)]

Anonymous (not verified) said: September 04, 2019

Permalink (/comment/283699#comment-283699)

> Some solutions, because of the way they were designed or coded, remove some fingerprinting vectors but introduce some artifacts or inconsistencies in the collected fingerprints.
>
> For example, imagine a browser extension that changes the value of fingerprints before they are sent.
>
> Everything works perfectly except the fact that the developer forgot to override the navigator.platform value. Because of this, the user-agent may say that the browser is running on Windows whereas the platform still indicates it is on a Linux system. This creates a fingerprint that is not supposed to exist in reality and, as such, make the user more visible online.

Yet that's precisely what Tor Browser has been doing for some time now, despite the strong backlash from users. Here's what AmIUnique says:

HTTP headers attributes
User agent: Mozilla/5.0 (Windows NT 6.1; rv:60.0) Gecko/20100101 Firefox/60.0

Javascript attributes
User agent: Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0
Platform: Linux x86_64

gk (/user/55) said: September 04, 2019

Permalink (/comment/283726#comment-283726)

That's true. However, the example was that some random user installs the extension and is now suddenly standing out even though they thought they would enhance their privacy. The Tor Browser case is different in that it is not just a single user behaving that way but all of the Tor Browser Linux users which should give cover against getting singled out.

Ideally, we would spood the JavaScript attributes as well, I agree. But there are usability concerns mostly on macOS that lead us to the current solution.

Anonymous (not verified) said: September 05, 2019

Permalink (/comment/283743#comment-283743)

Someone using Tor Browser is already willing to sacrifice usability for privacy, in many cases to much more extreme extent than not having websites detect their OS. For example, many websites become inaccessible, either the request is directly rejected or indirectly via infinite captchas. Having to manually choose or hunt down proper OS version when doing something OS specific is a minuscule issue in comparison, as well as much more infrequent for most users.

You could anonymize those Javascript attributes only at *Safer* security level where the user is more willing to sacrifice usability, or fully anonymize them at all security levels except for macOS builds, or a combination of both (*Standard* security level + macOS build = no anonymization).

gk (/user/55) said: September 05, 2019

Permalink (/comment/283794#comment-283794)

The security slider is for defenses against browser exploits: just for raising your *security* as a trade-off against functionality. We should not put *privacy* features in that mix as the result is hard to analyze and confusing to users. So, that's not a good option.

Anonymous (not verified) said: September 08, 2019

Permalink (/comment/283892#comment-283892)

I urge Tor Project to reassess these judgments in view of the latest revelations about dragnet attacks on a significant percentage of the (so far) living human population, this time apparently by China, together with the mounds of evidence that NSA is hardly a reformed character when it comes to their own vast "collect it all" dragnet surveillance programs.

Specifically, it should be clear that those users who tried to warn for years that everyone is a target have been correct all along. Which is obviously a crucial insight for making good decisions about trading off security viz. usability.

The most dangerous situations, wrt oppressive governments (and increasingly, they are all oppressive to one degree or another) arise when people falsely assume they enjoy protection, for example because they assume (falsely) that "ordinary citizens" are not targeted, or will not suffer dire consequences if a state-sponsored attack succeeds in trojaning their device.

Anonymous (not verified) said: September 08, 2019

Permalink (/comment/283879#comment-283879)

> Someone using Tor Browser is already willing to sacrifice usability for privacy, in many cases to much more extreme extent than not having websites detect their OS. For example, many websites become inaccessible, either the request is directly rejected or indirectly via infinite captchas.

You probably mean to say "Some Tor Browser users are willing...", but just to clear: I am one who thinks JS should be disabled by default and I tend to avoid sites which require it. If I do visit such a site, I use a different browser session. And I almost never try to use sites which demand that I fill in a captcha.

Anonymous (not verified) said: September 13, 2019

Permalink (/comment/284034#comment-284034)

"I am one who.." is rethinking what I wrote just above after reading what gk said above about the distinction between security enhanscements and privacy enhancements :-)

Anonymous (not verified) said: September 08, 2019

Permalink (/comment/283891#comment-283891)

Which begs the question: why not implement the JS spoofing for all versions of TB but the MacOS version?

If that is not practical, why not? And in that case, does the number of Mac OS users who use TB really justify the risk to everyone else?

Anonymous (not verified) said: September 17, 2019

Permalink (/comment/284079#comment-284079)

Then why not use the same UA ("Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0" in the previous post) both in the http header and in the JS attribute ?

siro (not verified) said: September 06, 2019

Permalink (/comment/283826#comment-283826)

The term "Linux system" is totally wrong, you should use the term GNU/Linux. GNU/Hurd based on Hurd, Hurd is based on Mach.

Anonymous (not verified) said: September 09, 2019

Permalink (/comment/283922#comment-283922)

Loyal to a fault, I grab the opportunity to thank Richard Stallman and Linus Torvalds for their positive contributions to the welfare of all humans.

Anonymous (not verified) said: September 04, 2019

Permalink (/comment/283701#comment-283701)

Why is Do Not Track not set in Tor Browser's HTTP headers by default?

gk (/user/55) said: September 04, 2019

Permalink (/comment/283727#comment-283727)

Because it's only saying "Dear website, please, please don't track me" where the website owner can still ignore that and track someone as they want. We want privacy by design where we don't have to beg anyone for that. Thus, Do Not Track is not helpful and we just don't use/send it.

Anonymous (not verified) said: September 08, 2019

Permalink (/comment/283893#comment-283893)

Since I am one of those who criticize certain design decisions by Tor devs, I should perhaps say that this is one I happen to agree with.

"Do not track" seemed like a good idea many years ago, but it has been clear from years that it cannot possibly ever work. And I'd suggest to anyone who wants to send the message to Silicon Valley that they desire or even demand that their privacy be respected, I think that's the right attitude, but is better achieved by using Tor Browser, which sends *two* messages: I demand that Silicon Valley respect my privacy, and I am sufficiently serious about that to use the best available privacy-enhancing tool currently available, which is Tor Browser.

ocho (not verified) said: September 04, 2019

Permalink (/comment/283715#comment-283715)

When I used Tor in 2017, Tor connects 3 hops that are in the same country. Can you fix it?

Anonymous (not verified) said: September 08, 2019

Permalink (/comment/283887#comment-283887)

Tor Project cannot fix anything after it has happened, unfortunately. But you probably meant to ask whether a problem you experienced in 2017 has been addressed, and to the best of my knowledge, a good deal of attention has been paid in recent years (owing in great part to vociferous user complaints!) to the issue of node geolocation, not just at the level of country but

more importantly to the problem of two or more nodes which are physically located in same server farm (for example a specific sever for hire facility in Amsterdam). AFAIK this remains a work in progress but I'd be happy to hear something from TP about the state of the art of node geolocation diversity.

Anonymous (not verified) said: September 09, 2019

Permalink (/comment/283929#comment-283929)

Node location is less concerning than if padding and timing of packets were not made to appear similar, but they are. And nodes are shared by many users simultaneously, and each domain you visit goes through a different circuit. Many defenses are working in concert. However, nodes could run compromised software that negates some defenses. The effect of those nodes could be suppressed, among other methods, by the directory servers doing more authentication of node software or by making sure country + AS (autonomous system) do not overlap for the three nodes chosen to build a circuit. But if you limit yourself from large groups of nodes as you build circuits, you affect other statistics of identifying your traffic. Tor benefits from more diversity in general. Nodes in particular are sparse in Asia, Africa and South America, and node operators can find a list of tor-friendly and unfriendly ISPs/CDNs (https://trac.torproject.org/projects/tor/wiki/doc/GoodBadISPs) on the Trac wiki.

good job *thumbs up (not verified) said: September 04, 2019

Permalink (/comment/283718#comment-283718)

Great article. Hopefully that will help the letterboxing naysayers understand a bit better.

Anonymous (not verified) said: September 08, 2019

Permalink (/comment/283895#comment-283895)

What means "letterboxing"?

Anonymous (not verified) said: September 09, 2019

Permalink (/comment/283920#comment-283920)

Letterboxing is black bars around video or images to fit in a different sized display. You will see it in Tor Browser soon because it helps to impede browser fingerprinting that detects your window resolution numbers.
https://en.wikipedia.org/wiki/Letterboxing_(filming) (https://en.wikipedia.org/wiki/Letterboxing_(filming))
https://en.wikipedia.org/wiki/Device_fingerprint (https://en.wikipedia.org/wiki/Device_fingerprint)
https://www.zdnet.com/article/firefox-to-add-tor-browser-anti-fingerpri… (https://www.zdnet.com/article/firefox-to-add-tor-browser-anti-fingerprinting-technique-called-letterboxing/)

Anonymous (not verified) said: September 13, 2019

Permalink (/comment/284030#comment-284030)

Did not know that; thank!

Anonymous (not verified) said: September 04, 2019

Permalink (/comment/283738#comment-283738)

> If you want to see your own browser fingerprint, I invite you to visit AmIUnique.org.
Ad? Do you want to see the crowds of crying "I'm unique!" users again?
My FP:
"All time : But only 3 browsers out of the 1263391 observed browsers (<0.01 %) have exactly the same fingerprint as yours."

Anonymous (not verified) said: September 07, 2019

Permalink (/comment/283852#comment-283852)

Me too (using Tails).

It seems a major reason may be that TB (in Tails at least) is *still* not properly choosing a standard size for the TB window

It seems a major reason may be that TB (in fails at least) is *still* not properly choosing a standard size for the TB window.

Results are worse if you enable JS (slider at "safer") but bad even if you put slider to "safest".

How to interpret this? I think questions about the authors tool should be addressed in comments or in a followup.

Plug-the-bug (not verified) said: September 05, 2019

Permalink (/comment/283768#comment-283768)

I was surprised that in TB8 dom.storage.enabled and browser.storage.enabled are set to true?
Firefox is a leaky boat and it seems some at Mozilla are working with a drill on new versions.

gk (/user/55) said: September 05, 2019

Permalink (/comment/283796#comment-283796)

We partition DOM storage so that it's not usable as a tracking mechanism more across different websites. Outright disabling a feature is just the last resort but luckily we can do better in that case.

Anonymous (not verified) said: September 12, 2019

Permalink (/comment/283996#comment-283996)

> Outright disabling a feature is just the last resort

Am I correct in guessing that your thinking here is that disabling a feature like DOM storage entirely would likely be noticeable by websites which could exploit this to more easily distinguish Tor Browser users from "ordinary FF"? But surely they can easily see from the IP that the visitor is coming from a Tor exit node?

Cannot Tor Project bring back Pierre Laperdrix for a followup explaining why he guesses Tor users are reporting the "almost unique" results from his fingerprinting test tool? I hope that part of the answer would be that the results reported by this tool are based almost entirely upon non-Tor users, but no-one has actually stated that, and I have found through long experience that bad things happen when no-one bothers to ask or answer questions about thoughtless assumptions which might prove to be very incorrect.

I think I support the general goal of making Tor users hard to distinguish from others (but only until almost everyone uses Tor for almost everything of course) while also making it hard to distinguish individual Tor users from other Tor users, and I can see that this hard. So we are asking questions not to criticize, only just to know.

gk (/user/55) said: September 13, 2019

Permalink (/comment/284019#comment-284019)

I am not concerned about "more easily" detecting Tor Browser users apart from Firefox users. There is probably no way to hide the former in the latter. The goal is to have a large as possible crowd of Tor Browser users being on the same Tor Browser version.

Disabling things like DOM storage harms that goal in that this breaks functionality that leads users away from Tor Browser.

Yes, Tor Browser users stand out compared to other browser users. That's already visible in the Panopticlick test which is often confusing to users. If one gets a lot of test results from browser users not using Tor Browser and mixes that up with Tor Browser user test results then it's expected that the latter stand out, which is not an issue as long as the group itself does not vary (much).

Not Spam (not verified) said: September 05, 2019

Permalink (/comment/283778#comment-283778)

Is there some reason that the Canvas Blocker extension is not installed by default? When it SHOULD be? It functions perfectly by default providing random hash codes for both DOMRect and canvas, leaving the user with total fingerprinting protection.

https://github.com/kkapsner/CanvasBlocker/ (https://github.com/kkapsner/CanvasBlocker/)

gk (/user/55) said: September 05, 2019

Permalink (/comment/283795#comment-283795)

It's not needed and adds additional risk to the browser as it is additional code that is running in it which would need to get audited (basically constantly to make sure no holes are introduced with new versions). We provide a proper defense by default in Tor Browser instead.

Not Spam (not verified) said: September 06, 2019

Permalink (/comment/283814#comment-283814)

https://www.browserleaks.com/canvas (https://www.browserleaks.com/canvas)

I've just enabled javascript for this site.

My fingerprint signature, after refreshing the page, remains static. This mean I have been positively identified by the hash code.

Can you please elaborate how I am protected against fingerprinting when I have just proven otherwise?

Thank you.

gk (/user/55) said: September 09, 2019

Permalink (/comment/283904#comment-283904)

All Tor Browser users are sending the same value back by default. See: https://2019.www.torproject.org/projects/torbrowser/design/ (https://2019.www.torproject.org/projects/torbrowser/design/) section 4.6. Cross-Origin Fingerprinting Unlinkability and there the HTML5 Canvas Image Extraction sub-section.

Anonymous (not verified) said: September 12, 2019

Permalink (/comment/283997#comment-283997)

It might be useful to list some of the things we might have in mind when we say that we Tor users want to appear "just like the others". Off the top of my head:

o making Tor users hard to distinguish (by studying packets from consumer device to ISP gateway) from other ISP customers who are not using Tor,

o making it hard (for that "global adversary" of ours) to tell that someone using a device which has been assigned a particular IP is using Tor at all (as I understand it, this currently seemingly impossible because we need to contact a directory authority to even join the Tor network, and we know our enemy monitors all connections to the DAs),

o making Tor packets (entry to middle node for example) hard to distinguish from other TSL bitstreams (as I understand it, there has been some progress here but the problem is not yet full solved),

o making it hard to distinguish a Tor user who is sharing a file with OnionShare from other Tor users,

o making it hard to distinguish a Tor user who is surfing to site X hard to distinguish (from src <-> entry) from one who is surfing to site Y,

o making Tor circuits being used to contact a Secure Drop hard to distinguish from "ordinary Tor circuits" (ideally from any of src <-> entry <-> middle <-> exit <-> intro <-> dst),

o making Tor users (after their traffic emerges from exit nodes) hard to distinguish from non-Tor users (currently seemingly impossible because websites can see the IP corresponds to a Tor exit node),

o making it hard to identify a Tor user even when connecting from a region with few entry nodes or with few Tor users,

o making Tor users hard to distinguish from other Tor users (after their traffic emerges from exit nodes).

(I would welcome any additions or corrections to this list.)

Some other legit goals which might sometimes be hard to reconcile with at least some of the above:

o making it easy for Tor Project (or even users?) to distinguish nodes whose operators are attempting MITM or logging traffic or otherwise doing things which resemble spookery or deeply unethical "research" on highly vulnerable humans who have not even been asked for consent from the friendly nodes,

o making it easy for friendly node operators (or Tor Project?) to learn about and correct any mis-configurations or unpatched security flaws on their node.

Anonymous (not verified) said: September 08, 2019

Permalink (/comment/283889#comment-283889)

Regarding canvas fingerprinting, some months ago I began to notice a weird icon appearing at many sites. Eventually someone told me this is the canvas icon and that it appears when a website is asking permission to fingerprint your browser. Reading between the lines of what you wrote, I guess FF does not ask permission, it just silently gives up the fingerprinting data, whereas TB asks the user for permission. But why on earth would a TB user say "yes"? Except by mistake? And what happens if the user fails to answer the question? After a timeout does TB assume that the user has given permission? I hope not, but I worry.

In any case, until your statement I had no reason to think TB was actually blocking the fingerprinting, although I hoped this was the case.

IMO, too often TB team makes too little effort to explain things, which causes unnecessary FUD among users (who far from being criticized for worrying about such things, should be *praised* for asking for answers). "Too little", that is, if we lived in an ideal world where Tor Project had more time to address things like user feedback. Which I admit TP mostly does not.

Still, fingerprinting seems like such a basic topic and is essential to protect against to have any chance of meeting the anonymity goals which are driving more and more ordinary people to try Tor Browser

"Ordinary people": as you no doubt know, Google Project Zero just published evidence that *all* a bad actor (said to likely be associated with a Chinese military intelligence service) attacked *all* iOS users visiting certain unnamed but "very popular" websites using sophisticated state sponsored malware. One of the targeted sites is said to be youtube.com. Apple admitted that this appears to be true, but rather horrifyingly appeared to suggest that because the presumed targets were Uyghurs, "ordinary people" need not worry. I suspect this assumption on the part of Apple is flat out wrong, and in any case I hate the suggestion that Uyghurs are not people too. Any comment?

Do you know whether Debian is addressing problems which could cause trouble for Tails as they work to release Tails 4.0.0? What about the battery API issue?

Anonymous (not verified) said: September 09, 2019

Permalink (/comment/283930#comment-283930)

Beginning in Firefox 58, the canvas icon and prompt in FF/TB controls extraction (https://bugzilla.mozilla.org/show_bug.cgi?id=967895) of images drawn on the canvas. The HTML5 canvas feature allows a webpage to draw or animate images. Some

pages draw features on the canvas that a user may want, but extraction of those images is different. If the user fails to answer, Mike Perry said in that Firefox bug report, "In Tor Browser, we have opted to have the canvas return white image data until the user has accepted a doorhanger UI that flips a site permission to either enable or permanently block canvas access from that site."

Why are you asking Tor Project about iOS and youtube? Tor Browser doesn't even support iOS. Rather than asking Tor Project, ask individuals. Why are you asking Tor Project about Debian's effect on Tails? Ask Tails' developers about that: Support (https://tails.boum.org/support/), Contact (https://tails.boum.org/about/contact/). Tails' documentation (https://tails.boum.org/doc/about/tor/#relationship) states, "Tails is a separate project made by a different group of people."

Anonymous (not verified) said: September 13, 2019

Permalink (/comment/284031#comment-284031)

Sorry for the confusion. I was writing quickly. Let me try again.

> "Ordinary people": as you no doubt know, Google Project Zero just published evidence that *all* a bad actor (said to likely be associated with a Chinese military intelligence service) attacked *all* iOS users visiting certain unnamed but "very popular" websites using sophisticated state sponsored malware.

Point 1: large classes of ordinary citizens not only *can* be targeted by state-sponsored malware, this is actually happening right now. Which debunks a false argument against using encryption, Tor, etc.

Point 2: large classes of ordinary citizens not only *can* be targeted by intelligence agencies which have chosen to "expend" valuable "zero-days" on unsophisticated targets, this is actually happening right now. Which debunks a false argument against using the best available defenses, such as Tails.

> One of the targeted sites is said to be youtube.com.

The revelation which shocked the security world is that everyone who visited youtube.com (and some other very popular sites) while using an iOS device may have been pwned by an intelligence agency (apparently the "E team" from the Chinese military, i.e. this was state sponsored attack using valuable zero-days, but the malware was not "A team" quality.

> Any comment?

Any further light which can be shed upon the affair is potentially valuable information to Tor users seeking to assess the dangers we face.

> Mike Perry said in that Firefox bug report, "In Tor Browser, we have opted to have the canvas return white image data until the user has accepted a doorhanger UI that flips a site permission to either enable or permanently block canvas access from that site."

Uhmm... in English?

Anonymous (not verified) said: September 13, 2019

Permalink (/comment/284036#comment-284036)

(Not the OP, but struggling to understand what TB does with canvas):

I take the point about not wanting to introduce more third party and possibly buggy code than needed, but do I understand what Mike Perry (as quoted elsewhere on this page) said to mean that when current TB sees a website asking for canvas data, it returns a blank white canvas image and puts up a weird little icon which is intended to warn the user that the site attempted canvas fingerprinting, and if the user clicks on the weird little icon which is intended to suggest a "canvas", the dialog they see means that TB is assuming they want to prevent that site from canvas fingerprinting the user, but they have the option to allow this if for some reason they want to allow it.

My horrible of expressing myself reflects my confusion...

While I have your attention, another issue which came up is that it is all too easy to accidently hit that tiny box which instantly maximizes the Tor Browser window--- game over. Would it be hard to simply disable that maximization box? I can see why a FF user might want to be able to maximize their browser with a click on a box, but surely not TB user would be want to do that on purpose?

gk (/user/55) said: September 17, 2019

Permalink (/comment/284075#comment-284075)

That notification box will be gone with Tor Browser 9.0 and letterboxing enabled.

Anonymous (not verified) said: September 21, 2019

Permalink (/comment/284144#comment-284144)

Are you saying NoScript and the other included extensions have been/are "audited"? Scroll through the source looking for suspicious URIs and hope to find none? Or something more?

GT (not verified) said: September 05, 2019

Permalink (/comment/283788#comment-283788)

Bonjour je suis nouveau je commence à connaître Tor, car j,ai vu un docu à la télé on suggère d'utiliser Protomail , Gnupg, encrypter , Jetsi.org au lieu de google utiliser DuckDuckgo ou Startpage et utilliser une antenne sur le toit pour internet gratuit .J',ai trouvé cela très intéressant comme info .Quelqu'un peut me confirmer cela et suggestion merci.

Anonymous (not verified) said: September 08, 2019

Permalink (/comment/283888#comment-283888)

Tor Browser makes it easy (maybe too easy) to get in the habit of searching Duckduckgo engine rather than Google search engine. If you download and install Tor Browser, in the location pane (where the url appears), try typing something which does not begin with http: or https: The browser interprets that as a search query and redirects it to Duckduckgo (by default) or to another search engine of your choosing, via tor circuits. I would advise still avoiding using Google even while using Tor Browser precisely because, as one of the papers cited in the blog post reports, Google is still by far the most determined user* of tracking technologies which unfortunately can in many cases be used to deanonymize tor users.

*In the commercial realm. NSA has not stopped piggybacking on synchronized cookies (mostly from Google or Facebook) to track individuals using tor. You may have read some months ago fake news widely reported by the mainstream media, claiming that NSA was abandoning its web and phone metadata dragnets. (Synchronized cookies are considered metadata at NSA.) In fact, a few weeks ago, NSA demanded that Congress reauthorize those programs, not just for another five years, but *indefinitely*. But mainstream media mostly ignored that inconvenient truth.

Anonymous (not verified) said: September 09, 2019

Permalink (/comment/283933#comment-283933)

It isn't called the "location pane". It's called the address bar.

Too easy? Every major browser searches from the address bar. Pick your poison: DNS logs or search engine logs. One of the reasons why Google is so popular is because many browsers come installed configured to send all of your search queries to Google. If people are surprised to see results from a search engine that proudly asserts it doesn't track users or sell profiles, I say good. It introduces people to competitors that side more closely with users, and it introduces people to the browser's preferences so they learn how to change the search engine.

Anonymous (not verified) said: September 13, 2019

Permalink (/comment/284033#comment-284033)

I guess I need to get out more :-) Almost all my recent experience is with Tails or Debian, using Tor Browser (not FF).

A Nonny (Non?) (not verified) said: September 05, 2019

Permalink (/comment/283792#comment-283792)

It is always wonderful to see researchers who take the time to inform users about the state of the art. Especially when the news is not bad!

Years ago I recall A. Narayan claimed (at his website) to be in the process of fingerprinting every author who ever posted anything to the web using stylometry. I wonder whether you know what the current status of that is? I hope that as with browser fingerprinting these claims have proven to be overstated.

Anonymous (not verified) said: September 07, 2019

Permalink (/comment/283851#comment-283851)

Arvind Narayanan co-authored (https://www.theregister.co.uk/2018/03/16/identifying_anonymous_programmers/) a research paper in 2018.

I think these are not related to Narayanan, but they look fairly recent:

"The field is dominated by A.I. techniques like neural networks and statistical pattern recognition.... The content of data has a high accuracy in authorship recognition (90%+ probability)."
https://www.whonix.org/wiki/Stylometry (https://www.whonix.org/wiki/Stylometry)

"Emma (http://emmaidentity.com/) needs only 5,000 words to learn each unique writing style. Emma proves to be 85% accurate in attributing authorship."

http://www.aicbt.com/authorship-attribution/online-software/ (http://www.aicbt.com/authorship-attribution/online-software/)

"Software systems such as Signature, JGAAP, stylo and Stylene for Dutch make its use increasingly practicable, even for the non-expert."
https://en.wikipedia.org/wiki/Stylometry#Current_research (https://en.wikipedia.org/wiki/Stylometry#Current_research)

"John Olsson, however, argues that although the concept of linguistic fingerprinting is attractive to law enforcement agencies, there is so far little hard evidence to support the notion."
https://en.wikipedia.org/wiki/Forensic_linguistics#Linguistic_fingerpri...
(https://en.wikipedia.org/wiki/Forensic_linguistics#Linguistic_fingerprinting)

A Nonny (Non?) (not verified) said: September 09, 2019

Permalink (/comment/283923#comment-283923)

Yes, that is who I meant. Sorry for the goof.

Oops, sorry (not verified) said: September 08, 2019

Permalink (/comment/283885#comment-283885)

Naranayan

Anonymous (not verified) said: September 08, 2019

Permalink (/comment/283896#comment-283896)

The name is Narayanan, A. Narayanan.
Anonymous (not verified) said: September 06, 2019

Permalink (/comment/283806#comment-283806)

How much anonymity can be expecetd with Tor Browser on the low security setting? Is it even worth using at this point, or can every PC be easily distinguished anyways?
Anonymous (not verified) said: September 07, 2019

Permalink (/comment/283848#comment-283848)

> my Linux laptop
> the user wants to receive her page

Are you trying to tell us something, Pierre? (:

But seriously, the extent to which everyone is lulled into giving up privacy for convenience or bureaucratic expectation is horrifying. I'm very thankful the W3C stepped in front of development to write a fingerprinting guidance document. Now we need to pressure developers to adopt it before those developers release new RFCs or proofs of concepts.

This is the first time I'm hearing about FP Central. Great news. Did you and Tor Project collaborate with ghacksuserjs who commented in June (https://blog.torproject.org/comment/282858#comment-282858) about developing TorZillaPrint?

> some elements could be rendered improperly, they could be positioned at the wrong location

Didn't Acid3 take care of this?
Anonymous (not verified) said: September 08, 2019

Permalink (/comment/283878#comment-283878)

@TB team:

It seems that the TB team has abandoned some apparently useful things with no explanation, which I find frustrating as a user. One of these is relevant to the subject of this post:

Some time ago the TB team said TB would try to find a reasonable default size to avoid making users easily trackable because their screen size was essentially unique among Tor users owing to vagaries of the device on which they run TB.

But recently I noticed that the standard sizes seem to have been abandoned, and when I checked the amiunique website this appears to confirm that very weird nonstandard window sizes are being used to fingerprint TB users.

Any comment?

1 (/BROWSER-FINGERPRINTING-INTRODUCTION-AND-CHALLENGES-AHEAD?PAGE=0) /
2 (/BROWSER-FINGERPRINTING-INTRODUCTION-AND-CHALLENGES-AHEAD?PAGE=1) /
NEXT › (/BROWSER-FINGERPRINTING-INTRODUCTION-AND-CHALLENGES-AHEAD?PAGE=1)

Upcoming Events

November 06, 2019
Internet e anonimato: la rete Tor (Firenze) (/events/internet-e-anonimato-la-rete-tor-firenze)

November 08, 2019
Tor Meetup (Brussels) (/events/tor-meetup-brussels)

November 09, 2019 - November 10, 2019
Freedom not Fear (Brussels) (/events/freedom-not-fear-brussels)

November 11, 2019 - November 14, 2019
CriptoDunas (Fortaleza, Brazil) (/events/criptodunas-fortaleza-brazil)

November 15, 2019
The Future of Speech Online (D.C.) (/events/future-speech-online-dc)

See All Upcoming Events (/events/month)

Recent Updates

New Release: Tor Browser 9.0.1 (/new-release-tor-browser-901)

by boklm (/users/boklm) | November 05, 2019

Tor Browser 9.0.1 is now available from the Tor Browser download page (https://www.torproject.org/download/) and also from our (https://www.torproject.org/dist/torbrowser/9.0.1/</body></html>)

A better internet is possible. I've seen it. (/better-internet-possible-ive-seen-it)

by isabela (/users/isabela) | November 04, 2019

Surveillance, censorship, and tracking run rampant online.

Take Back the Internet with Us (/take-back-internet-us)

by Sarah (/users/Sarah) | October 28, 2019

You understand the importance of online privacy.

New Alpha Release: Tor 0.4.2.3-alpha (/new-alpha-release-tor-0423-alpha)

by nickm (/users/nickm) | October 24, 2019

There's a new alpha release available for download. If you build Tor from source, you can download the source code for 0.4.2.3-alpha from the download page (https://www.torproject.org/download/tor/) on the website. Packages should be available over the coming weeks, with a new alpha Tor Browser release in a couple of weeks.

Remember, this is an alpha release: you should only run this if you'd like to find and report more bugs than usual.

This release fixes several bugs from the previous alpha release, and from earlier versions of Tor.

Changes in version 0.4.2.3-alpha - 2019-10-24

- Major bugfixes (relay):
  - Relays now respect their AccountingMax bandwidth again. When relays entered "soft" hibernation (which typically starts when we've hit 90% of our AccountingMax), we had stopped checking whether we should enter hard hibernation. Soft hibernation refuses new connections and new circuits, but the existing circuits can continue, meaning that relays could have exceeded their configured AccountingMax. Fixes bug 32108 (https://bugs.torproject.org/32108); bugfix on 0.4.0.1-alpha.

- Major bugfixes (v3 onion services):
  - Onion services now always use the exact number of intro points configured with the HiddenServiceNumIntroductionPoints option (or fewer if nodes are excluded). Before, a service could sometimes pick more intro points than configured. Fixes bug 31548 (https://bugs.torproject.org/31548); bugfix on 0.3.2.1-alpha.