# Browser Fingerprinting in Tor Browser

Heeraj H Nair
Amrita School Of Engineering
Amritapuri
Email: heerajhnair@am.students.amrita.edu

***Abstract*—In this proposal, We investigate the extent of device fingerprinting possible in Tor browser via the different JavaScript API. We even examine how much perlavent are the older fingerprinting technique. We will also examine all the Javascript API's for finding new fingerprinting technique. We will try to produce a unique hash with the help of these results which will help the users to track without the help of cookies. We will be analyzing the Tor browser bug trac issues and some side channel attacks. After the end of the research we will come with add-on which will defend with the privacy issues in firefox. We will also propose our idea on how the defence can be made to defend this issue for Tor browser.**

***Keywords*—Tor, Fingerprinting, Tracking**

## I. Introduction

Web advertisers and analytic collect large volume of data based on our Internet browsing pattern. These type of data help them to make user profile which eventually reveals your user location, incomes, family status, interests and much about you. Identification of user can lead to showing higher prices, blocking some services or data and even directing to misleading information.

Most of the users to defend tracking usually delete their cookies but the tracking technologies are much evolved practice of deleting cookies have become less effective to users who is defending using deleting cookies. Many defencing tactics have been evolved in the recent years such as 'incognito' mode or private browsing, use of VPN or IP masking techniques, use of extensions and adds-on such as Ghostery, Disconnect, HTTPS everywhere etc.

But most of the defensive technique fail if the advertiser use the method of fingerprinting. Browser fingerprinting is a technique of uniquely identifying the individual user by the unique pattern of information collected which often includes browser type and version, operating system and version, screen resolution, font detection, plugins, time zones, languages, battery usage, hardware configurations and many more.

More that 2 Million of daily users use anonymous services such as Tor to defend client's privacy from his destination. Tor communicated through multiple number of relays using numerous number of encryptions such that none of the relays will be revealed the client's identity and destination. But still the advertisers can still cause privacy issues by using Browser fingerprinting.Tor browser has done series of patches to enhanse their privacy, they are using a component Tor button which takes care of Application level security and privacy concerns in Firefox. To protect against Tor exit Node eavesdropper, it includes HTTPS-Everywhere and to protect from optional defense against javascript and other potential exploit vectors using NoScript.

Tor browser doesn't support some APIs such as Battery API, Network Connection API and plugins such as Flash. Fingerprinting techniques like Font Fingerprinting are defended by disabling all the attempts to enumerate the system font by limiting the number of fonts in the system. Moreover the image data from the canvas is returned blank to defend against canvas Fingerprinting technique. But some of the browser properties has to released that are given same values and the properties that uncover the identity are simply disabled.

According to 2015 study, 90% .onion sites are distinguishable from the regular sites. We will also be dealing with side channel attacks, side channel attacks are based on the information gained from the physical implementation rather than brute force or theoretical weakness.

## II. Background and Related work

### Background

Krishnamurthy and Wills on their paper, Privacy Diffusion on the web[2] have done a longitudinal study where they have shown that tracking on the web have steadily increased and more an more techniques are formed in the coming years which implies there should be more study done on the area of privacy.

Privacy techniques has quite evolved in the past 10 years. Ten years back, for tracking the users HTTP cookies were used then it went towards Flash cookies. With the evolution of web and introduction of HTML5 the trackers or the third party advertisers started using HTML5 storage such as localStorage, sessionStorage, indexedDB and ETag for tracking users. But the limitation with the HTTP cookie user delete cookies within 30 days.

With the advancement of web tracking third party trackers and advertisers started focusing on browser fingerprinting. Browser fingerprinting is one of most used way of tracking user's nowadays. Many of the API such as battery API, network API, WebGL, Canvas, browser properties, plugins and many other propeties are being used for uniquely identifying the users. Beyond that there are even some research where the login leaks can identify which social network the users are currently logged in. Nowadays extensions and adds on have become a part of web browser, by loading the URI schema and event handlers we can make user profile which make the advertiser know which extensions are being used by a user.

Many of the browser fingerprinting techniques such as canvas fingerprinting, font and plugin enumeration, extension detection and other tracking issues which are still prevalent

in browsers such as Mozilla, chrome etc. But these issues have defense in the case of tor browser. the Some of side channels attacks and some other fingerprinting techniques such as os fingerprinting using math function, getClientRects fingerprinting and audio fingerprinting are still present. Canvas Fingerprinting[1] which one of the major fingerprinting techniques which is been used by big companies has defense technique with Tor Browser but it doesn't automatically deny canvas Fingerprinting or block it. It gives a prompt or user a access to whether allow it or not.

## III. Design and Model foI for Tor Browser

Tor browser is based on Mozilla Extended Support Release(ESR) Firefox branch. But there are many patches been applied to enhance the security and privacy. Some of the famous security features include HTTPS everywhere, which automatically switches thousands of sites from HTTP to HTTPS. Many of the patches are even done in Firefox and even the Tor browser features is augmented through the Tor browser extension. As a part of protection against Tor Exit Node eavesdroping, they have used HTTPS-Everywhere and to provide optional defense against javascript they have user NoSript and even they have made many modification in several of Firefox settings.

Privacy requirements help in reducing the privacy risk and ability of user's activity of one website to be linked with the user's activity of the another website without any knowledge of the user.

- Cross-Origin Identifier Unlinkability : Users activity must not be linked from one origin by any other origin without user interation and approval. This is mainly due to authentication tokens, browser identifiers and shared state.

- Cross-Origin Fingerprinting Unlinkability : Users activity must not be linked from one origin to another origin by any of the third party from fingerprinting browser behaviour

- Long-Term Unlinkability : browser should clear all its linkable state such authentication tokens and browser state and obtain a fresh identity.

Tor browser is not using any addons such as Addblock Plus, Ghostery, Priv3 which are actually avoided because they believe these are not going to ensure any privacy protection, Tor believe the development should be in general solution. (Give the dom change paper xhound) One of the research which happened in recent years showed that both extension and addons usage can be another fingerprinting vector as they can identifed using chrom URI and they produce change in the DOM which can lead us to easily who which extension they are using.

There are 3 diffginerent security level in Tor browser - high, medium and low. In High Javascript is disabled by default. In some sites Javascript is enabled performance optimizations are disabled. Scripts on some sites may run slower.HTML5 video and audio media become click to play via NoScript. Some mechanism of displaying math equations, some font rendering features, some types of images are disabled. In medium, same things are implied but Javascript is only disabled in non-HTTPS sites. Low level is most usable level in Tor browser. It doesn't block Javascript.

In Torbrowser the Firefox proxy settings use Tor directly as a SOCKS proxy. To prevent WebRTC at the compile time it is disabled. All the plugins such as flash are disabled in Tor browser. Firefox addons can perform arbitary activity including bypassing Tor, that is reason there is addons whitelist. It doesn't even load any system wide extensions[7]. Tor browser have all information written inside the bundle directory. Tor even disables third party cookies and all the cache enteries are isolated from URL bar domain. Similar to canvas can be used for Fingerprinting the WebGL is also used for fingerprinting, there was recent research which even suggested that WebGL can be even used for Cross Browser Fingerprinting. The defense stratergies used by Tor browser helps us to protect from the issue of privacy.

Operating system fingerprinting is not considered a big problem in privacy even though tor browser has reduced the entropy of finding the os to 2 bits. At least 3 HTML5 API Battery API, the Network Connection API and sensor API are disabled in the current browser because of fingerprinting. Even the mozilla services are disabled in Tor to make non of the mozilla services are contacted. Some of the feature in javascript functions like readers view and connection state has been disabled.

## Related Work

One of the recent paper from KU Leuven university[5] talks about some attacks in Tor browser which can deanonymize the user in Tor browser which can find out the IP address of the user. One of the attack mentioned by them was Cross Session Events in which if an devicechange event happen it would fire in all browsing context even in incognito mode, which can help in identifying and is an privacy issue. As a consequence of this the specification was altered to only fire the event on tabs that were either previously given permission to access media devices, or that are active. Otherwise this can even result in Cross browser fingerprinting. Similarly online/offline, languagechange, devicemotion, Battery had also some privacy issues of Cross Session Events. But some of these issues such as online/offline changes is already given protection in Tor browser. Some side channel issues such as the convert channel issue are also discussed in the research. Some of the website such as browser leaks which can be used for understanding the known techniques of fingerprinting in different browsers.

## Previous findings

We were looking to into bugtraq of Tor browser we were able to find different ways of fingerprinting Tor browser. Some of the Javascript API which were responsible for fingerprinting were removed from the Tor browser but inspite of that there are some API which is needed for the proper functioning of website. For example Math in Javascript, which is an important function in the case of web which will be used by many developers. With the help of Math one can do OS fingerprinting. If the value of Math.tan(-1e300) is -1.4214488238747245 then the os it is Linux machine and if the value is then -4.987183803371025 then it is Windows machine.

Some of the low hanging fingerprinting targets would be the User-agent, Accept-* headers, pipeline usage, request orderings and what are the custom filters which are being used

for privacy protection. Javascript Date object reveals OS type can be used for OS Fingerprinting, the result or the date format returned is different for different OS.

JavaScript API can be used get the size of the screen and window even though some efforts are been made by Tor browser to give protection. Till now correct patch regarding the screen width is not available. Canvas fingerprinting have certain defense in the case of Tor browser but actually in spite of forcefully stopping it, it asks the user as a prompt whether they wanted to block the canvas fingerprinting or not.

Font fingerprinting can be even done in the current version of Tor browser but they have reduced the fingerprinting effect by reducing the number of fonts being used in the system. CSS media queries can be used by the third party trackers for getting more information about the desktop size, widget size, display type, DPI, user agent type about the user which was formerly more available to JavaScript.

Website traffic fingerprinting is an attempt to recognize the encrypted traffic patterns of specific websites, this may take in place between user and guard node. There has been many researches and papers in this area providing defenses and protection for example: using random padding in CSS, Js and other files by using comments which will make attacker job of fingerprinting website very difficult.

| Techniques | Defense(tb) | Privacy in others |
|---|---|---|
| Canvas Fingerprinting | Yes | No |
| WebGL Fingerprinting | Yes | No |
| Extension detection | Yes | No |
| Font Enumeration | Partial | No |
| OS Fingerprinting | No | No |
| Simultaneous event | No | No |
| Convert Channel | No | No |

## IV. Timeline

We will be testing all the issues which are present in the Tor bugtrac. Mostly the open issues will be given more priority. We will be testing the results in different operating system such as Linux, Windows and Mac Os. We will be also be evaluating the privacy concerns of all the JavaScript API's about the 742 API's listed in the w3c for finding the privacy issues in the Tor browser.

Next step would be using the data for making a hash which could help in identifying the user's uniquely. Already there are many known methods for OS fingerprinting and even for Tor browser version fingerprinting and tails version fingerprinting. We will be also using some of the methods which has privacy issues in Tor browser for fingerprinting.

Last part of the research is the defense, we will behave to blacklist or remove some of the JavaScript API. We will also have to choose methods of padding or some way of confusing the third parties such that more false positive arise such that extent of fingerprinting goes down.

## V. Conclusion

Our work was intended to improve the security of Tor browsers. Tor browser is regarded as the most secure and more private browser, for improving the privacy more researches has to be done in the browser fingerprinting part. In this paper we have intended to show the issues with JavaScript based web fingerprinting in Tor browsers and its implications.

## References


[1] Mowery, K., Shacham, H.: Pixel perfect: Fingerprinting canvas in HTML5 In: Fredrikson, M. (ed.) Proceedings of W2SP 2012. IEEE Computer Society (May 2012), https://cseweb.ucsd.edu/hovav/dist/canvas.pdf

[2] KRISHNAMURTHY, B., AND WILLS, C. Privacy diffusion on the web: a longitudinal perspective. In Proceedings of the 18th international conference on World wide web (2009), ACM, pp. 541 550

[3] Design and Implementation of Tor Browser. https://www.torproject.org/projects/torbrowser/design/

[4] BOJINOV, H., MICHALEVSKY, Y., NAKIBLY, G., AND BONEH, D. Mobile device identification via sensor fingerprinting. arXiv preprint arXiv:1408.1416 (2014).

[5] Tom Van Goethem, Wouter Joosen, One Side-Channel to Bring Them All and in the Darkness Bind Them: Associating Isolated Browsing Sessions, woot17, 2017

[6] Get the Tor browser version and tails version, https://github.com/jonaslejon/tor-fingerprint/

[7] Oleksii Starov and Nick Nikiforakis. XHOUND: Quantifying the Fingerprintability of Browser Extensions. In *Proceedings* of ACM, 2013.