Uncovering Canvas Fingerprinting in Real-Time and Analyzing Its Usage for Web Tracking

A study of Web tracking techniques used by the 500 most popular websites of Germany.

Philip Raschke[1], Axel Küpper[2]

Abstract: Since the beginnings of the World Wide Web, advertisements are an inherent part of it. It enables business models that allow users to consume content and services free of charge, while their providers generate revenue with every visitor. These advertisements are specifically tailored to each user to increase the probability that a user actually clicks on them. However, in order to personalize these advertisements, tracking techniques are used to generate profiles for every visitor. Canvas fingerprinting is a modern technique to identify users across multiple websites. Since the generation of such fingerprints takes place completely absent from the users' attention, this paper presents a browser extension to uncover these fingerprints in real-time. Furthermore, an analysis of today's usage of canvas fingerprinting is conducted to evaluate the role of it in Web tracking practices. We found that canvas fingerprinting is hardly used for Web tracking, while simultaneously learning that it is applied in the field of cyber security.

Keywords: Targeted advertising, canvas fingerprinting, data privacy, data protection, Web security

1 Introduction

Targeted advertising and behavioral targeting in particular are popular techniques, which are widely used by websites to generate revenue to compensate for freely provided content. Advertisements are indispensable from an economical perspective considering the costs involved in the production of the respective content and the maintenance of an ideally scalable infrastructure to serve numerous users with the content in question. Nowadays, many business models rely on targeted advertising and behavioral targeting. Services like Google, Facebook and others would not be free of charge without advertisements. Thus, the public benefits from these techniques by enabling the realization of those business models.

[1] Technische Universität Berlin, Service-centric Networking, Ernst-Reuter-Platz 7, 10587 Berlin, philip.raschke@tu-berlin.de

[2] Technische Universität Berlin, Service-centric Networking, Ernst-Reuter-Platz 7, 10587 Berlin, axel.kuepper@tu-berlin.de

However, there is another side of the coin. To tailor advertisements specifically for each user based on her or his activity requires technologies to monitor and track users across multiple websites, to extract and derive information from this behavior, and to generate an ideally clear profile that can be exchanged with other third parties.

There are numerous tools and browser extensions that promise to protect users' data privacy by the attempt to prevent tracking across websites, yet providers of these tools play a cat-and-mouse game with the advertising networks, which constantly develop new techniques to circumvent these tools. Moreover, despite an increasing group of users, who use tools to block advertisements, these tools are still used by a minority of users. A fact that advertisers learned to benefit from: the detection of the usage of such tools is considered in the generated profile. This way privacy-aware users can be easily targeted.

Even more severe is the fact that most users are completely unaware of being subject to tracking while browsing the Web [TH15]. Most users are not aware of involved third parties, which may be located abroad, when visiting the website of a content provider. This circumstance is in strong conflict with the General Data Protection Regulation[3] (GDPR), which came into effect in May 2018. According to the GDPR, personal data must be processed *"in a transparent manner"*[4], hence users must be aware of being tracked while browsing the Web.

While deleting cookies on a frequent basis and blocking third party cookies in general might help to protect against Web tracking to some extent, it does not protect against modern tracking techniques like *canvas fingerprinting*. The canvas element was introduced with version five of the Hypertext Markup Language (HTML5) enabling websites to animate and render two-dimensional graphics on the screen. Researchers found in 2012 that each machine renders an image slightly different, consequently a fingerprint can be derived that is highly unique [MO12]. Since then, canvas fingerprinting has been reportedly used to track users.

Detecting a website using canvas fingerprinting is not trivial. For this reason, this paper develops a tool, which detects the usage of the canvas element and persistently logs generated images, which can be later manually assessed to determine whether a website used the canvas for legitimate purposes or for tracking. This way, the paper aims to identify websites that actually use canvas fingerprinting and further to visualize the images used to generate the fingerprints. After the development of the above described tool, the 500 most popular websites of Germany will be tested to assess the adaption of canvas fingerprinting by websites.

[3] Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation), OJ L 119, 4.5.2016, p. 1-88 [hereinafter GDPR]

[4] GDPR art. 5(1)(a)

The remainder of this paper is structured as follows: Section 2 presents related work in this field summarizing the most important contributions of other researchers. Followed by Section 3, which presents the developed prototype and the methodology to detect the usage of the canvas element. In Section 4, the evaluation will be presented and finally Section 5 concludes the contribution of this paper.

2 Related work

Web tracking techniques have a long history and are almost as old as the World Wide Web itself. An extensive study conducted by Lerner et al. [LE16] investigates used tracking techniques present in archived websites between 1996 and 2016. The authors demonstrate the increased number of trackers, who compete against each other to sell the most detailed user profiles to their customers. From this perspective canvas fingerprinting is a promising technology to gain a competitive advantage over other trackers. With the release of version 58, Firefox started to warn users when websites attempt to use the canvas feature and explicitly ask users for permission beforehand [EX17]. Unfortunately, the warning message lacks giving information on what canvas fingerprinting is, involved risks, and the consequences of a decision. A more sophisticated approach to circumvent canvas fingerprinting is offered by the browser extension *Canvas Defender* [CA17], which is available for Chrome and Firefox.

Canvas fingerprinting is a rather recent technique, which was first presented by Mowery and Shacham [MO12] in 2012. Their work demonstrates how the back then new HTML5 feature could be used to generate a relatively unique fingerprint, that can be used to track users. Bujlow et al. [BU17] published in 2017 a comprehensive survey on Web tracking techniques, including canvas fingerprinting. The authors of this paper classify canvas fingerprinting as *browser instance fingerprinting*, since the resulting fingerprint may differ from one browser to another on the same machine.

To assess the popularity of canvas fingerprinting as Web tracking technique, Acar et al. [AC14] investigated the globally 100.000 most popular websites according to Alexa.com [AL18]. Their study revealed that a rather small group of about 5.5% of the websites actually use canvas fingerprinting to track users. However, the vast majority of fingerprints are produced by a script of a single third party.

The same group continued their research and extended the sample size resulting in the to date most sophisticated analysis of used Web tracking techniques. In their publication, Engelhardt et al. [EN16] present *OpenWPM*, which is a framework to detect and analyze most known tracking techniques. In their paper of 2016, they analyzed one million websites. They found that an even tinier fraction of websites (just 1.6%) use canvas fingerprinting for user tracking. Their work confirms that the majority of fingerprints is produced by a small number of third party scripts.

There are various tools and browser extensions that identify those mostly obfuscated third-party scripts and prevent their execution. But as stated above, the providers of such tools constantly struggle to keep their software up to date to correctly identify the tracking third parties. A study by Merzdovnik et al. [ME17] evaluated the effectiveness of these tracker blocking tools. To achieve this, they also used the top 100.000 Alexa.com websites and measured how well the different blocking tools protected them from known trackers. Their findings suggest that the application of these tools and browser extensions helps to reduce the extent of tracking the user is subject to when browsing the Web but does not prevent it entirely.

Sanchez-Rola et al. [SA17] discuss in their publication further countermeasures against Web tracking. They, alternatively to the rather conventional approach to detect and block tracking third parties, propose further anti-tracking techniques such as user profile spoofing and dedicated software like specific browsers. They further emphasize the role of standardization and regulation to tackle the issue of Web tracking. However, they neglect or underestimate the role of transparency as a possible countermeasure to Web tracking. Informing users about applied tracking techniques at the very moment it happens enables them to discontinue browsing the respective website or behaving accordingly to the situation.

3 Prototype

The main challenge with regard to the actual implementation is to detect and log generated canvas fingerprints in real-time while the user browses a website. This challenge was approached by researchers differently. Le et al. [LE17] modified the source code of the Firefox browser to accurately detect obfuscated Web tracking scripts of third parties. However, this is no option for our work, since a modification to the browser would require users to use a dedicated version of their browser that needs to be maintained in parallel to the development of Firefox. Ikram et al. [IK16] propose a machine-learning approach to classify third-party JavaScript programs. This approach is well-suited for a real-time analysis of tracking activities. However, it does not enable the visualization of canvas fingerprints that can be shown to the user.

3.1 Canvas fingerprint detection

Our prototype aims to inject program logic that detects the usage of the canvas element and the call of functions that are usually used to generate the fingerprint. This implies that legitimate usage of the canvas feature is also detected and would be falsely marked as canvas fingerprint. Thus, a manual revision of the detected fingerprints is necessary to exclude false positives. It is also thinkable to offer users a feedback mechanism to mark false positives, which will be ignored the next time the users visit the respective website. The generated and logged images could be also analyzed to automatedly detect similarities between fingerprints. This way, regular images could be also excluded right away. However, the handling of false positives is not within the scope of this paper.

The main methodology to detect generated canvas fingerprints relies on the possibility to override the JavaScript application programming interface (API). This way, code can be injected to monitor, analyze, and even alter the behavior of included JavaScript programs. To inject code on every website, our prototype is realized as browser extension for the browser Chrome. This extension inserts a piece of code on every website ideally as first script in the document. Consequently, all following scripts use the modified JavaScript functionality. To detect the functions that have to be overridden, we refer to the already presented work of Bujlow et al. [BU17], who identified that the functions *getImageData* and *toDataURL* are used to extract image information from the canvas. In addition to these two functions, we learned that the function *toBlob* [HT18] could be used for this purpose as well by taking a look at the browser extension Canvas Defender, which protects against canvas fingerprinting by adding persistent noise to the generated images. Our modification to these three functions consists of the logging and the writing to the local storage of the results of these functions.

JavaScript interface	**Method**
Document	createElement
	createElementNS
	getElementById
	getElementsByName
	getElementsByClassName
	getElementsByTagName
	getElementsByTagNameNS
HTMLCanvasElement	toBlob
	toDataURL
CanvasRenderingContext2D	getImageData

Table 1: Complete list of overridden JavaScript APIs.

Unfortunately, overriding these three functions is not sufficient, since most canvas fingerprints are generated within *iframe* elements, which have their own scope and thus are not affected by the modifications to the JavaScript API. Fortunately, to execute JavaScript within these frames, their sandbox attribute needs to be set to *"allow-scripts"* in order to generate the canvas fingerprints, which enables us to inject our script into these frames as well. This requires the detection of iframe elements present in the document or the detection of JavaScript programs that create an iframe dynamically. Consequently, further JavaScript functions have to be overridden. The identification of these functions was also taken from the Canvas Defender browser extension. See Table 1 for a complete list of overridden JavaScript functions.

3.2 Presentation of the fingerprints to the user

In order to actually provide transparency to the user, the logged canvas fingerprints need to be shown to the user. To achieve this, on every website our browser extension inserts a small panel that can be shown and hidden by a simple click and which contains the generated images of the corresponding website. See Figure 1 for a screenshot of the browser extension in use.

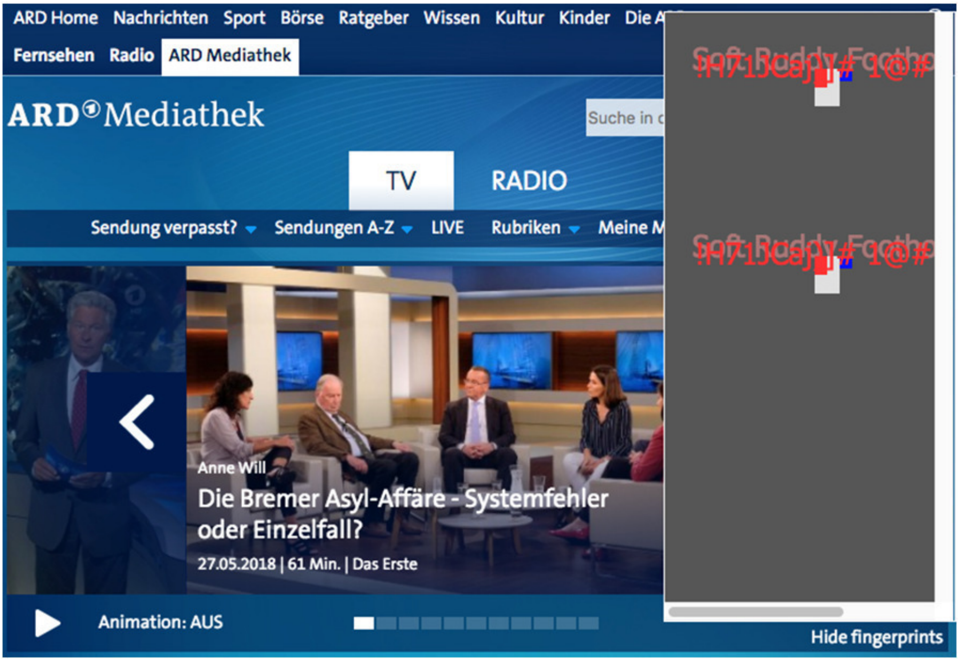


Figure 1: Visualization of canvas fingerprints detected on a website.

To preserve the user experience, the panel is hidden by default and can be optionally displayed when the user wants to. The choice of the background color of the panel has rather practical reasons, since some elements of the fingerprints are not visible on a pure white or pure black background. As it can be also seen in Figure , fingerprints are often captured twice, since they are generated with different functions. In case of the situation displayed in the screenshot, the function *getImageData* and *toDataURL* are used, thus the same fingerprint is logged twice. This is supposedly done to increase the entropy of the generated canvas fingerprints. A further improvement of the browser extension for future work is to detect duplicates, so that there are not shown to the user in order to prevent possible confusion.

4 Evaluation

The second aim of this paper is to investigate on today's usage of canvas fingerprinting as Web tracking technique. Therefore, the 500 most popular websites of Germany (according to Alexa.com) are examined to see which websites generate canvas fingerprints or which websites use services of third parties that involve canvas fingerprinting. Based on the findings of related work presented in Section 2 of this paper, we expect a rather small number of websites that use canvas fingerprinting for Web tracking.

To conduct the experiment, we used the browser automation software Selenium [SE17] and its Python implementation that besides automation further allows detailed browser configuration. This is useful to determine the state of the browser when running the experiment. In our case it is important to ensure that there are no cookies, no entries in the browser history, or any data in the local storage of the browser.

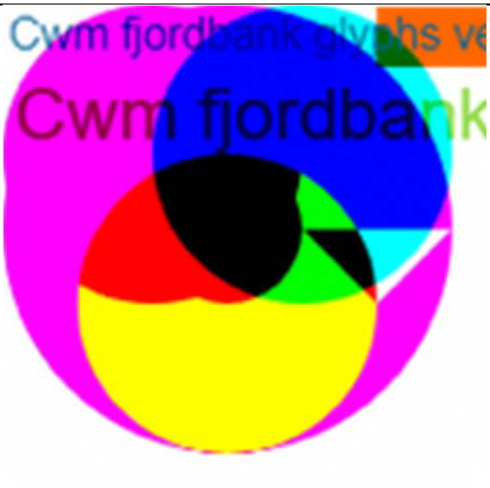
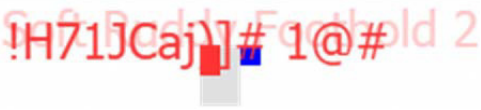
The evaluation was conducted on a single Windows machine with an Intel Core i7-7700HQ and 16GB memory. Equipped with this computational power, there was no need for distributed computing technologies, since visiting 500 pages is rather feasible on a single machine.

4.1 Results

Our experiment identified 22 websites (4.4%) out of the 500 websites, which generated canvas fingerprints. In addition to these 22 websites, 15 further images were logged that were not generated for Web tracking purposes. The process to determine the purpose of the generated image was conducted manually. Images that were used within the website and displayed as content have been classified as regular use of the canvas feature. In theory these images could be used for fingerprinting as well, however it is rather unlikely, since fingerprints use certain techniques (such as texts, shadows, and overlaying elements) to maximize entropy. See Table 2 for an overview of the captured canvas fingerprints and the websites that use them. Note that the images have been cropped for visibility reasons.

As it can be seen, we could identify only seven distinct images as canvas fingerprints. The most frequently appearing fingerprint (FP1) is used by more than the half (12 websites) of websites that use canvas fingerprinting. The origin of this specific canvas fingerprint is the open source library *fingerprintjs2* [VA18], which is publicly available on GitHub. Furthermore, fingerprint FP3 has the same origin but is produced by an earlier version [VA17] of the library. The fingerprints FP4 and FP5 are generated by scripts of the respective websites, while the fingerprints FP6 and FP7 are generated by third party scripts.

Remarkable is the origin of fingerprint FP2, which is generated by a script that is hosted on the corresponding websites accessible via a dynamically changing path and is not served by a third party. The path always includes a directory called *akam*. The fingerprint is only generated during the first visit and is not produced afterwards. Only a reset of the browser, i.e. deleting cookies, the browser history, and data in the local storage leads to a new production of the fingerprint when revisiting the website. The JavaScript that generates the fingerprint is highly obfuscated, so no insights can be gained by investigating it. This fingerprint is clearly not used for Web tracking. Research revealed that the content delivery network and cloud solutions provider Akamai [AK18], uses the canvas fingerprint most likely in one of its products [BO18] to detect malicious bots.

ID	Fingerprint	Websites
FP1		• dastelefonbuch.de • de.nametests.com • de.xhamsterlive.com • gidonline.in • kinogo.cc • livejasmin.com • notebooksbilliger.de • poppen.de • txxx.com • upornia.com • vk.com • yourporn.sexy
FP2		• ardmediathek.de • ikea.com • mobile.de • motor-talk.de

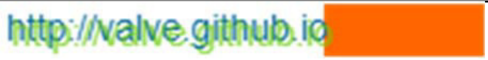
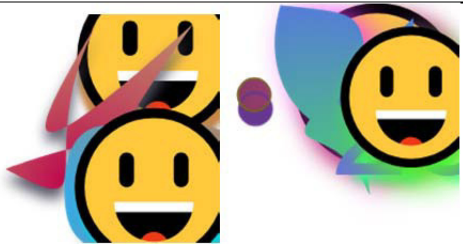
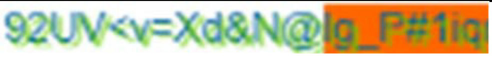
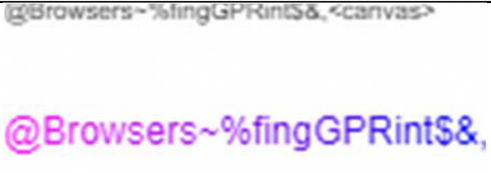
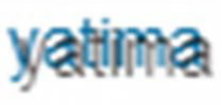
ID	Fingerprint	Websites
FP3		- om.forgeofempires.com - zalando.de
FP4		- facebook.com
FP5		- linkedin.com
FP6	 @Browsers~%fingGPRint$&,<canvas> @Browsers~%fingGPRint$&,	- airbnb.de
FP7		- netzwelt.de

Table 2: Overview of logged fingerprints and the websites that use them.

4.2 Summary and discussion

The results of the evaluation are in accordance with the results of the presented related work: just a tiny fraction of websites actually uses canvas fingerprinting to track users across websites.

A possible reason for this relatively low application of canvas fingerprinting in the field of Web tracking is that the technology is still rather recent and lacks of many implementations. The results show that most websites use a single open source library to realize canvas fingerprinting. The computational overhead involved in generating the canvas fingerprints might also lead to the low adaption rate of canvas fingerprinting. Its advantage over conventional tracking techniques is also debatable.

However, the results also show that other potential application scenarios for canvas fingerprinting can be found in Web security products to detect malicious bots or even fraud. We incidentally discovered that a German bank uses canvas fingerprinting for their online banking service probably to detect or even prevent potential fraud. It could be used as evidence to proof that a certain machine visited a website, when the owner of the machine claims the opposite or the vice versa.

5 Conclusion

This paper developed a browser extension to uncover canvas fingerprints that are often generated absent from the awareness of the user. It is questionable whether this practice is compliant with new legal requirements imposed by the GDPR. Our browser extension, in contrast to other privacy-enhancing tools, provides transparency to users by informing them when a website uses canvas fingerprinting to track their behavior. While it is also possible to alter the fingerprints in a way that there are useless for Web tracking, we decide to emphasize transparency and inform users on solutions to circumvent canvas fingerprinting. An analysis has been carried out to assess how canvas fingerprinting is used today and to which extent today's users are subject to it. This analysis shows that canvas fingerprinting is only used by a small group of websites and also for other purposes than Web tracking.

In following research, it would be interesting to see how users react to the notice, which informs them about canvas fingerprinting while viewing a certain website. It is questionable whether users change their behavior after being informed that they are observed. It must be further evaluated whether users benefit from the visualization of the canvas fingerprints or not. It is also reasonable to frequently monitor the adoption rate of canvas fingerprinting within the scope of Web tracking to identify an increasing or decreasing usage. Our prototype will be extended by detecting further browser fingerprinting methods and other tracking techniques to provide more transparency to users.

6 Acknowledgments

Supported by the European Union's Horizon 2020 research and innovation programme under grant 731601.

Literature

[AC14] Acar, G. et al.: The Web Never Forgets: Persistent Tracking Mechanisms in the Wild. In: Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security - CCS '14. 674–689, 2014.

[AK18] Akamai | CDN und Cloud-Lösungs-Anbieter. https://www.akamai.com/de/de/, last accessed: 05/28/2018.

[AL18] Alexa - Website Traffic, Statistics and Analytics. https://www.alexa.com/siteinfo, last accessed: 05/28/2018.

[BO18] Bot Manager | Product Briefs | Akamai. https://www.akamai.com/us/en/multimedia/documents/product-brief/bot-manager-product-brief.pdf, last accessed: 05/28/2018.

[BU17] Bujlow, T. et al.: A Survey on Web Tracking: Mechanisms, Implications, and Defenses. In: Proceedings of the IEEE. 1476–1510, 2017.

[CA17] Canvas Defender - Chrome Web Store. https://chrome.google.com/webstore/detail/canvas-defender/obdbgnebcljmgkoljcdddaopadkifnpm, last accessed: 05/28/2018.

[EN16] Englehardt, S., Narayanan, A.: Online Tracking: A 1-million-site Measurement and Analysis. In: Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security - CCS'16. 1388–1401, 2016.

[EX17] Extensions in Firefox 58 | Mozilla Add-ons Blog. https://blog.mozilla.org/addons/2017/11/20/extensions-in-firefox-58/, last accessed 06/23/2018.

[HT18] HTMLCanvasElement.toBlob() - Web APIs | MDN. https://developer.mozilla.org/en-US/docs/Web/API/HTMLCanvasElement/toBlob, last accessed: 05/28/2018.

[IK16] Ikram, M. et al.: Towards Seamless Tracking-Free Web: Improved Detection of Trackers via One-class Learning. 2016.

[LE16] Lerner, A. et al.: Internet Jones and the Raiders of the Lost Trackers: An Archaeological Study of Web Tracking from 1996 to 2016. In: Proceedings of the Usenix Security Symposium 2016. Texas, USA. 2016.

[LE17] Le, H. et al.: Towards accurate detection of obfuscated web tracking. In: Proceedings of IEEE International Workshop on Measurement and Networking, M and N 2017. 1–6, 2017.

[ME17] Merzdovnik, G. et al.: Block Me if You Can: A Large-Scale Study of Tracker-Blocking Tools. In: Proceedings - 2nd IEEE European Symposium on Security and Privacy, EuroS and P 2017. 319–333, 2017.

[MO12] Mowery, K.; Shacham, H.: Pixel Perfect: Fingerprinting Canvas in HTML5. In: Proceedings of Web 2.0 Security & Privacy 2012 (W2SP). San Francisco, USA. 1–12, 2012.

[SA17] Sanchez-Rola, I. et al.: The web is watching you: A comprehensive review of web-tracking techniques and countermeasures. Logic Journal of the IGPL 15/2017. 18–29, 2017.

[SE17] Selenium - Web Browser Automation. https://www.seleniumhq.org/, last accessed: 05/28/2018.

[TH15] Thode, W.; Griesbaum, J.; Mandl, T.: "I would have never allowed it": User Perception of Third-party Tracking and Implications for Display Advertising. In: Proceedings of the 14th International Symposium on Information Science (ISI 2015). Zadar, Croatia. 445-456, 2015.

[VA17] Valve/fingerprintjs: Anonymous browser fingerprint. https://github.com/Valve /fingerprintjs, last accessed: 05/28/2018.

[VA18] Valve/fingerprintjs2: Modern & flexible browser fingerprinting library. https://github.com/Valve/fingerprintjs2, last accessed: 05/28/2018.