Web Browser Data Collection and User Awareness Regarding Web Browser Data Exposure

Luka Hrgarek, Tatjana Welzer, Marko Hölbl
Faculty of Electrical Engineering and Computer Science
University of Maribor
Smetanova ulica 17, 2000 Maribor, Slovenia
{luka.hrgarek, tatjana.welzer, marko.holbl}@um.si

Abstract. *The World Wide Web has been transformed from a collection of hyperlink-connected documents to a global application platform. During this transformation process the role of users has changed from passive readers to contributors, collaborators and interactive users who have made their (personal) data a valuable resource for web applications. We present the possibilities and methods of browsers and user devices data collection via a dedicated web application and the analysis of user awareness regarding the possibility of collecting such data. The results show that web applications can retrieve a large amount of data about browsers and devices without the user's permission. Furthermore, our research showed that users are well aware of data collection possibilities.*

Keywords. privacy, web browser, fingerprinting

1 Introduction

In the early age of the Web, users enjoyed a large level of anonymity (Puglisi, Rebollo-Monedero, & Forné, 2015). In 1990, the Web was just a collection of static hypertextual documents connected by hyperlinks (Berners-Lee & Cailliau, 1990) – no user input or personalization was available. Nowadays, when taking into account all the changes which the web has undergone, many users are confronted daily with the problem of privacy while using the web. Websites and applications record users' data, which is for most users out of their scope of awareness. Such monitoring of user browsing habits raises privacy concerns since the collected data can be used to build a user profile, which is especially worrying if such a profile can be linked with the user's identity (Wills & Zeljkovic, 2011).

The degree of privacy, which websites offer to their users can be closely related to the amount of user trust. McKnight and Chervany (1996) defined *trust* as "*the extent to which one party is willing to depend on somebody or something in a given situation with a feeling of relative security, even though negative consequences are possible*". If a website does not offer adequate privacy to users, this greatly impacts their trust towards a website.

In this paper we study the extent to which data can be obtained from user devices using web browsers, together with techniques and methods of how to implement this collection.

2 Data collection methods

When users browse the Web, a complex network of personalization services monitor their preferences via the tracking of their browsing habits. This data is used to provide tailored suggestions, in terms of products users could buy, interesting resources, social connections, etc. Personalization services rely on combining different services and techniques to track users across different websites and applications (Puglisi et al., 2015).

We will describe different techniques, which make obtaining data about user browsers and devices possible.

A **web cookie** is a small string (usually just an ID-number) that a server sends to the user's web browser. The cookie is saved on the user's hard drive and later sent back to the server. Browsers started using cookies in 1995 to help facilitate authorized user access and personalization settings. Web cookies have two major drawbacks: the cookie can identify only one browser application and it can be deleted, which means that the identifier can be lost (Boda, Földes, Gulyás, & Imre, 2011). However, the method is still widely used (Gomez, Pinnick, & Soltani, 2009).

Capturing browsing history is a technique, which exploits the JavaScript method `getComputedStyle()` that can check the color of a hyperlink. In a hypothetical scenario, a website can automatically generate an array of hyperlinks as DOM objects and execute `getComputedStyle()` on each hyperlink. In this way, a website can determine which sites you have already visited and make a profile of your interests. As of 2010, CSS 2.1 specifications (World Wide Web Consortium et al., 2011) include a note in which W3C warns about the aforementioned shortcomings. Also, Weinberg, Chen, Jayaraman, and Jackson (2011) conclude that capturing and analyzing browsing history causes privacy and security risks which outweigh potential benefits. Today it is no longer possible to capture browser history with the `getComputedStyle()` method.

Geolocation API, defined by W3C, is a high-level interface to location information of a device, such as latitude and longitude. The API is designed to allow *one-shot* and continuous access to location data. However, specifications do not guarantee that the interface will provide the exact location. Since W3C considers that location data can potentially compromise the user's privacy, the browser vendor's implementation of the specification must *provide a mechanism that protects the user's privacy* and *should ensure that no location information is made available through this API without the user's express permission* (Popescu, 2013).

Battery Status API is a browser mechanism that enables access to power-related data. It can provide data about the battery level, the charging time, the discharging time and the charging status (Lamouri & Kostiainen, 2016). In JavaScript, it is accessible via the `navigator.getBattery()` method. The API does not require explicit user permission to access this data, which means that any website or related third party script can use it. Likewise, the specifications do not require web browsers manufacturers to implement a mechanism that would inform the user about the website accessing their battery data (Diaz, Olejnik, Acar, & Casteluccia, 2015). In the "*Security and privacy considerations*" section of the W3C specification that describes the Battery Status API, the following is stated: "*The information disclosed has minimal impact on privacy or fingerprinting, and therefore is exposed without permission grants*" (Lamouri & Kostiainen, 2016). Nevertheless, Diaz et al. (2015) showed that the Battery API, as implemented by GNU/Linux and Firefox browser, enables device fingerprinting and tracking due to high precision measurements.

Device Orientation API allows for the obtaining of information about the physical orientation and movement of the hosting device. Mobile devices, such as mobile phones, tablets and smart watches use this data to automatically rotate the display to remain in an upright position. The interface for the orientation of the device in a W3C-specification was introduced in March 2010. The first browser which partially implemented this feature was Google Chrome version 7 in October 2010 (Can I use, n.d.). The W3C considers that this information is not sensitive enough for a need to request the user's authorisation to operate. However, recent academic efforts showed how device orientation sensors can be used for device and browser fingerprint. Despite W3C's assertions that access to device orientation sensors is not safety-critical, Mehrnezhad et al. (2016) in his study showed the potential risks. By analyzing and processing the input data using neural networks, it was possible to guess a four-digit PIN code with a success rate of 74%. In the second and third attempt, the success rate of identifying increased to 86% and 94%, which represents a serious threat. The mentioned findings facilitated the idea of possible user permission requirements or a visual indicator which would inform the user about the sensor's usage.

Ambient Light Sensor provides information on the amount of light in the room where the device is located. The main measurement used built-in light detector device and the results is the unit *lux* (Kostiainen & Langel, 2016). In the "*Security and privacy considerations*" section of the W3C specification that describes the Ambient Light Sensor indicate that there are no specific security and privacy considerations regarding the use of an ambient light sensor. However, Azizyan, Constandache, and Roy Choudhury (2009) have shown that the ambient light sensor in combination with other mobile sensors, such as sound and geolocation, provides a platform of interoperable and mergeable data that can be used to derive a logical location.

User-Agent is a field contained in the HTTP request header. It can be accessed via the JavaScript method `navigator.userAgent` which returns a string like `Mozilla/5.0 (X11; Linux x86 64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/57.0.2987.133 Safari/537.36`. The user-agent can be accessed without permission and contains data about the operating system, the browser vendor and version, the rendering engine, browser details etc.

3 Web application and survey

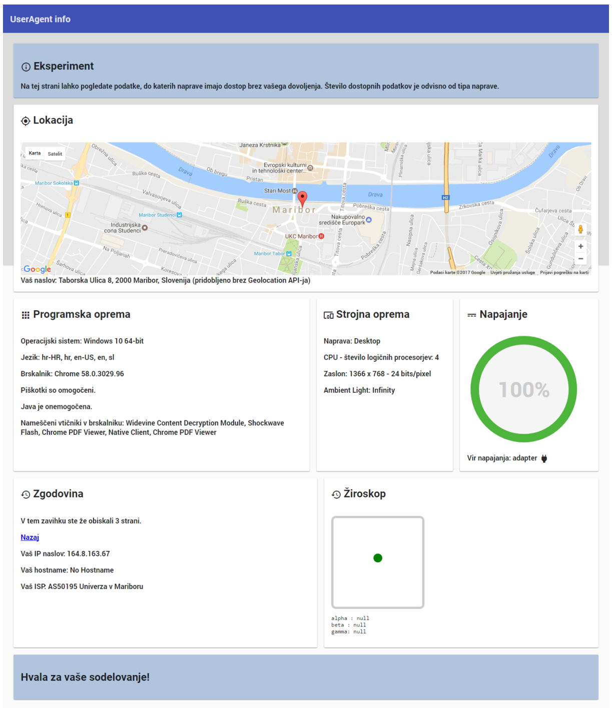
In scope of the possibilities of browser fingerprinting of user data, we conducted research into the possibilities of data collection using web browsers. Additionally, we conducted a survey to investigate the scope of user awareness regarding the possibilities and threats of web browser data collection.

3.1 Web application

We developed a web application with the help of which we analyzed the possibilities of access to browser and device data by a website. The obtained data were divided into two parts. In the first part we asked the users for their permission to access their camera and geolocation API. The second part (Figure 1) of the application captured a range of data which is accessible without any user interaction, i.e. without their permission or awareness. The application backend was developed using the Node.js framework, a single-page frontend with AngularJS and the CouchDB database for data storage.

Each entry in database contained:

- A timestamp,
- UUID identificator,



UserAgent info

ⓘ Eksperiment

Na tej strani lahko pogledate podatke, do katerih naprave imajo dostop brez vašega dovoljenja. Število dostopnih podatkov je odvisno od tipa naprave.

Lokacija

Karta
Satelit
Drava
Stari Most
Maribor
UKC Maribor
Nakupovalno središče Europark
Evropski kulturni in tehnološki center...
Industrijska cona Studenci
Google
Podaci karte ©2017 Google
Uvjeti pružanja usluge
Prijavi pogrešku na karti

Vaš naslov: Taborska Ulica 8, 2000 Maribor, Slovenija (pridobljeno brez Geolocation API-ja)

Programska oprema

Operacijski sistem: Windows 10 64-bit

Jezik: hr-HR, hr, en-US, en, sl

Brskalnik: Chrome 58.0.3029.96

Piškotki so omogočeni.

Java je onemogočena.

Nameščeni vtičniki v brskalniku: Widevine Content Decryption Module, Shockwave Flash, Chrome PDF Viewer, Native Client, Chrome PDF Viewer

Strojna oprema

Naprava: Desktop

CPU - število logičnih procesorjev: 4

Zaslon: 1366 x 768 - 24 bits/pixel

Ambient Light: Infinity

Napajanje

Vir napajanja: adapter

Zgodovina

V tem zavihku ste že obiskali 3 strani.

Nazaj

Vaš IP naslov: 164.8.163.67

Vaš hostname: No Hostname

Vaš ISP: AS50195 Univerza v Mariboru

Žiroskop

```
alpha : null
beta : null
gamma: null
```

Hvala za vaše sodelovanje!

Figure 1. The developed web application for browser data collection

- camera permissions (`true/false`),
- possible camera error description,
- location permission (`true/false`),
- possible location error description,
- operating system,
- primary and supported languages,
- browser name and version,
- cookies enabled (`true/false`),
- Java enabled (`true/false`),
- number of logical processors,
- screen resolution,
- number of pages visited in current tab,
- IP address.

The database stored all the data with the aforementioned structure in the JSON format.

Some methods of data collection in browsers today are no longer available, because the W3C pointed to potentially dangerous in their requirements and recommendations functionality. An example is the exploitation of the functionality of the pseudo-class `a:visited`, to determine which sites a user has visited without his consent.

All data cannot be retrieved on all devices, since there are a variety of hardware and software limitations. The web application has been tested on devices running Microsoft Windows 10, Ubuntu Linux 16.04, Android 7.1.2, iOS 10.3 and Windows Phone 8.1. Difficulties occurred when displaying web pages in Microsoft Edge on the Microsoft Windows Phone 8.1, in which the web page was not displayed correctly. Mozilla Firefox supports most of the functionality of the web application with the exception of access to the Battery Status API and Device Orientation API. In Safari on iOS it is not possible to access the camera and battery status. Although the feature is implemented in Safari's engine, Apple has not enabled it (Olejnik, 2016). Opera supports all the functionality of the web application, since it uses the same rendering engine as Google Chrome.

3.2 Survey

To assess the privacy and data exposure awareness of web users, a survey was conducted in May 2017. The survey was distributed among the students and staff of the Faculty of Electrical Engineering and Computer Science at the University of Maribor and one high school.

The survey consisted of 32 questions that were divided into 3 parts. After the first part with demographic questions, the respondents were provided with 6 claims relating to the user's online experience, awareness of privacy threats and social networks. Next to each claim a 5-point Likert scale was offered. We also wanted to know if the user was willing to allow a website to access their location and camera. The third part of the survey started with the following claim: *I believe that it is possible that a website can access the following data about my device without my permission.* Hereinafter were listed 17 different possibilities and respondents were asked to answer on a 5-point Likert scale with an additional possibility of *I do not know this concept*.

At the end of the survey, for the purpose of tracking a particular respondent between the survey and web application, we generated a UUID number for each respondent which was linked with the survey and concatenated with the web application URL as a GET parameter.

3.3 Results and findings

The data from the survey and the web application was collected for 14 days. In that time 108 respondents participated, of which only 76 completed the survey and used the web application.

Among the respondents there were 27 women (36%) and 49 men (64%). Most of the respondents were in the age group 18-24 years (68%), and the age group 34-44 years (16%). Other age groups were represented with 9% or less.

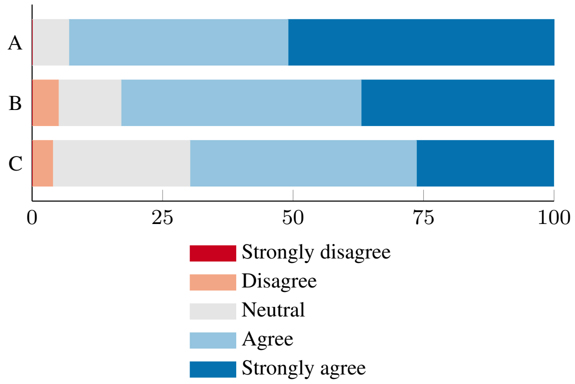


Figure 2. The ratio between the answers to the questions *How much do you agree with the following statement? – I have a lot of experience using the Web* (A), *I believe that I am aware of online threats* (B) and *Anonymity on the Web is important to me* (C)

Most respondents thought that they were very experienced web users, since 51% of them fully agreed (choice 5) and 42% agreed (choice 4) with the statement (Figure 2, A). They also thought that they were well aware of the dangers and threats on the web (Figure 2, B).

Users were most aware of the possibility that a website could access data about the type of their device (mobile/desktop), their operating system and their language without their permission. Only 28% of the respondents allowed access to their location and only 14% of respondents allowed access to their camera. Despite the fact that the majority of respondents believe that they are experienced and informed, it turns out that this does not affect their awareness of how much it is possible to obtain information about their browser.

4 Conclusion

The use of different Web applications, which often acquire the personal data of users is increasing (Krishnamurthy & Wills, 2006). However, each individual can decide whether they will provide data to an application, web site, provider or not. On the other hand, web applications are able to access certain data using browsers without user permission and awareness. Due to today's relationship between users and their devices (computers, smartphones,...) and installed browsers, the question of data access using web browsers is today more relevant than ever before. It is possible to locate, identify and follow every device that has Internet access (Michael & Clarke, 2013). Therefore, they can be used for monitoring and tracking, in a manner that adversely affects the interests of the user.

An example of the (mis)use of this data is the case of the American company Uber. It was uncovered that Uber charged people with low or almost empty batteries more, since they were more likely to pay for surge (Keith Chen, 2016). This further raised concerns that other companies could also abuse such data (e.g. analyze the battery's status in order to increase the prices for users) (Biz Carson, 2016; Jordan Golson, 2016).

We do not claim that all collectable data is a threat to privacy. However, it is important that users are aware of *what* is collected, *how* and whether this is *really necessary*.

References

Azizyan, M., Constandache, I., & Roy Choudhury, R. (2009). Surroundsense: mobile phone localization via ambience fingerprinting. In *Proceedings of the 15th annual international conference on mobile computing and networking* (pp. 261–272).

Berners-Lee, T., & Cailliau, R. (1990). Worldwideweb: Proposal for a hypertext project. *Retrieved on February*, *26*, 2008.

Biz Carson. (2016). *You're more likely to order a pricey Uber ride if your phone is about to die.* `http://uk.businessinsider.com/people-with-low-phone-batteries-more-likely-to-accept-uber-surge-pricing-2016-5`.

Boda, K., Földes, Á. M., Gulyás, G. G., & Imre, S. (2011). User tracking on the web via cross-browser fingerprinting. In *Nordic conference on secure it systems* (pp. 31–46).

Can I use. (n.d.). *Can I use DeviceOrientation.* `https://caniuse.com/#search=deviceorientation`. (Accessed on 2017-05-13)

Diaz, C., Olejnik, L., Acar, G., & Casteluccia, C. (2015). The leaking battery: A privacy analysis of the html5 battery status api. In *Lecture notes in computer science* (Vol. 9481, pp. 254–263).

Gomez, J., Pinnick, T., & Soltani, A. (2009). Knowprivacy. *School of Information.*

Jordan Golson. (2016). *Uber knows you'll probably pay surge pricing if your battery is about to die.* `http://www.theverge.com/2016/5/20/11721890/uber-surge-pricing-low-battery`.

Keith Chen. (2016). *This Is Your Brain On Uber.* `http://www.npr.org/2016/05/17/478266839/this-is-your-brain-on-uber`.

Kostiainen, A., & Langel, T. (2016, August). *Ambient light sensor* (W3C Working Draft). W3C. (https://www.w3.org/TR/2016/WD-ambient-light-20160830/)

Krishnamurthy, B., & Wills, C. E. (2006). Generating a privacy footprint on the internet. In *Proceedings of the 6th acm sigcomm conference on internet measurement* (pp. 65–70).

Lamouri, M., & Kostiainen, A. (2016, July). *Battery status API* (Candidate Recommendation). W3C. (http://www.w3.org/TR/2016/CR-battery-status-20160707/)

McKnight, D. H., & Chervany, N. L. (1996). The meanings of trust.

Mehrnezhad, M., Toreini, E., Shahandashti, S. F., & Hao, F. (2016). Stealing pins via mobile sensors: Actual risk versus user perception. *arXiv preprint arXiv:1605.05549*.

Michael, K., & Clarke, R. (2013). Location and tracking of mobile devices: Überveillance stalks the streets. *Computer Law & Security Review*, *29*(3), 216–228.

Olejnik, L. (2016). *Browsers remove functionality due to privacy.* `https://blog.lukaszolejnik.com/browsers-remove-functionality-due-to-privacy/`. (Accessed on 2017-09-05)

Popescu, A. (2013, October). *Geolocation API specification* (W3C Recommendation). W3C. (http://www.w3.org/TR/2013/REC-geolocation-API-20131024/)

Puglisi, S., Rebollo-Monedero, D., & Forné, J. (2015). You never surf alone. ubiquitous tracking of users' browsing habits. In *International workshop on data privacy management* (pp. 273–280).

Weinberg, Z., Chen, E. Y., Jayaraman, P. R., & Jackson, C. (2011). I still know what you visited last summer: Leaking browsing history via user interaction and side channel attacks. In *Security and privacy (sp), 2011 ieee symposium on* (pp. 147–161).

Wills, C. E., & Zeljkovic, M. (2011). A personalized approach to web privacy: awareness, attitudes and actions. *Information Management & Computer Security*, *19*(1), 53–73.

World Wide Web Consortium, et al. (2011). Cascading style sheets level 2 revision 1 (css 2.1) specification. `https://www.w3.org/TR/CSS2/selector.html#link-pseudo-classes`.