Browser fingerprint and privacy issues regarding data collection from search engines*

Viktória Latičová
Slovenská technická univerzita v Bratislave
Fakulta informatiky a informačných technológií
xlaticova@stuba.sk

12. december 2023

Abstract

This article aims to reflect on some key privacy concerns caused by search engines collecting data while searching on the internet. Besides the obvious task of searching by keywords, they also collect user data to create detailed profiles. Browser fingerprinting techniques identify users uniquely, allowing cross-site tracking leading to location tracking, privacy invasion, or third-party sharing. Protecting users' privacy requires not only technological solutions-because completely removing our browser fingerprint would make searching impossible, but also legal regulations and raising public awareness and education about this topic.

1 Introduction

Tracking users across various websites is not a novel concept in the online realm. Websites often monitor users without their awareness, and these practices can serve various purposes. Some are well-intentioned, such as personalization and user interface optimization, while others involve the sale of user information to third-party websites or the more extensive tracking of personal data.

2 browser fingerprint

The most well-known tracking technique to the public is collecting cookies, which are then stored on users' devices for later use [14]. But unlike cookies, creating a browser fingerprint actually does not require the permission of the user and as it is not stored in any device, it does not leave a trace either. Tracking via cookies also became harder as modern browser extensions automatically delete them after a certain time.

A browser fingerprint is essentially a user profile created from data collected during online activities [11]. This profile includes a wide range of device-related information, such as IP address, time zone, CPU details, screen resolution, plugins, ad-blockers, and much more. Some webmail services are even recognized for inspecting emails of users, who have never granted permission to them. [2] Their unique fingerprint is now a new tool for websites to assign their identity to activities on the internet.

[*] Semestrálny projekt v predmete Metódy inžinierskej práce, ak. rok 2023/24, vedenie: Ing. Mohammad Yusuf Momand, MSc.

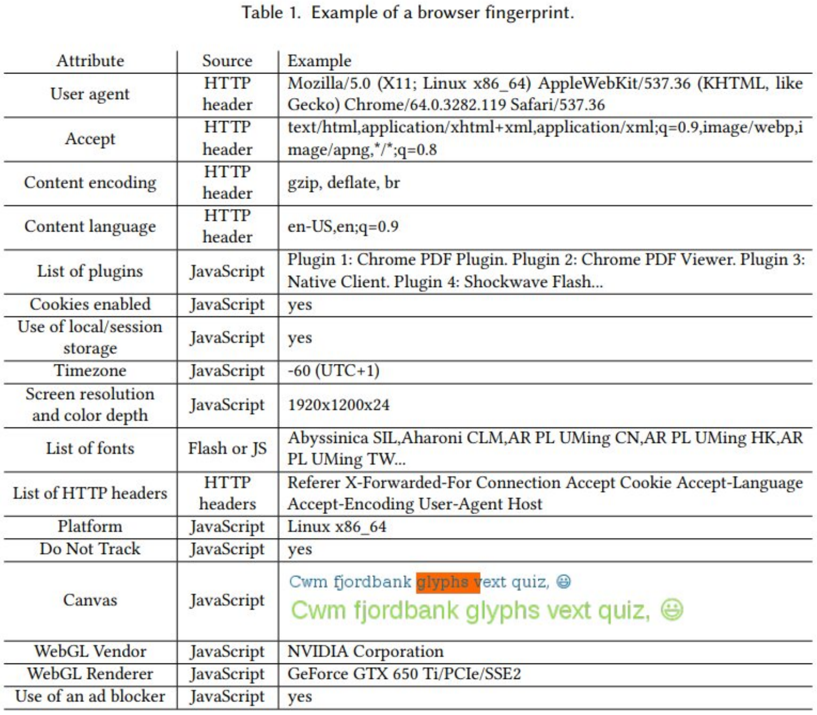


Table 1. Example of a browser fingerprint.

Attribute	Source	Example
User agent	HTTP header	Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/64.0.3282.119 Safari/537.36
Accept	HTTP header	text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8
Content encoding	HTTP header	gzip, deflate, br
Content language	HTTP header	en-US,en;q=0.9
List of plugins	JavaScript	Plugin 1: Chrome PDF Plugin. Plugin 2: Chrome PDF Viewer. Plugin 3: Native Client. Plugin 4: Shockwave Flash...
Cookies enabled	JavaScript	yes
Use of local/session storage	JavaScript	yes
Timezone	JavaScript	-60 (UTC+1)
Screen resolution and color depth	JavaScript	1920x1200x24
List of fonts	Flash or JS	Abyssinica SIL,Aharoni CLM,AR PL UMing CN,AR PL UMing HK,AR PL UMing TW...
List of HTTP headers	HTTP headers	Referer X-Forwarded-For Connection Accept Cookie Accept-Language Accept-Encoding User-Agent Host
Platform	JavaScript	Linux x86_64
Do Not Track	JavaScript	yes
Canvas	JavaScript	Cwm fjordbank glyphs vext quiz, 😃 Cwm fjordbank glyphs vext quiz, 😃
WebGL Vendor	JavaScript	NVIDIA Corporation
WebGL Renderer	JavaScript	GeForce GTX 650 Ti/PCIe/SSE2
Use of an ad blocker	JavaScript	yes

Figure 1: Example of a browser fingerprint. [8](Laperdrix et al., 2019)

2.1 click tracking

One of the first methods, but still used for obtaining information from users is quite straightforward – using URLs [2]. Many people remain unaware that major search engines like Google, Yahoo, or Bing permit this kind of click tracking [16]. Tracking clicks on links is not always employed for bad purposes. In most cases, it is used in affiliate marketing by companies for statistical and analytical purposes, and to better understand the user's behavior.

In certain cases, attackers can create a unique link that initially redirects the user to a different site, takes information that is needed, and swiftly redirects the user to the destination site. This method can be dangerous because of its almost non-existent chance of being discovered just by looking at the URL.

2.2 geolocation fingerprinting

One of the most straightforward features to identify is the IP address of a user, However, obtaining a user's location from an IP is a more complicated process, known as geolocation. When a user accesses a website, their IP address is captured. IP geolocation databases of providers can provide an approximate geographic location. This information can include the country, city, or even latitude and longitude. The presence of a proxy can be easily detected and, with some skill, even bypassed [2].

At the same time, it is important to note that geolocation is not always 100% accurate. It only provides an approximation of location. Also, a VPN or a Tor (The Onion Router) can be used to mask the real IP address.

2.3 operating system fingerprinting

Operating system fingerprinting is a technique used for detecting the operating system running on a remote device when connected to a network. Based on an article [1] of Aksoy et al. OS detection helps improve network management, ensures that the software is compatible with the user's operating system, and also can detect unauthorized attacks. Javascript and Flash have the ability to detect information about operating systems of devices [2]. Information like local timezone is detectable in a blink of an eye. According to Bujlow et al. [2], Flash has the ability to detect various system features, such as audio capabilities, camera and microphone access permissions, and printing support.

We can categorize OS fingerprinting into two types: [5]

passive fingerprinting, where the attacker does not send out any packets to the victims. In this method, the attacker does not directly communicate with the target system but rather waits for the target to establish a connection, most of the time by being tricked to connect.

Active fingerprinting, on the other hand, requires the attacker to send specific network requests to the victim in order to be able to determine system information.

In summary, passive fingerprinting relies on the victim's system making the first move, while active fingerprinting requires the attacker to actively send requests to the victim's system to discover its operating system.

2.4 browser extensions fingerprinting

Browser extensions are programs, that are written to add functionalities to the browser. They offer users the ability to block ads, take screenshots, manage passwords, or assist when researching on the internet [6]. using more extensions extends the tracking powers of attacker, as users become more and more unique for fingerprinting. Some extensions, without the knowledge of the user, can be also designed to collect data from the user from the start [7]. These extensions are written using HTML, CSS, and JavaScript, which gives them the capability to track user activity similarly to websites [11]. The research of Oleksii Starov and Nick Nikiforakis [13] from the year 2017 identified that: „6.3% of popular Google Chrome extensions leak privacy-sensitive information to at least one-third party".

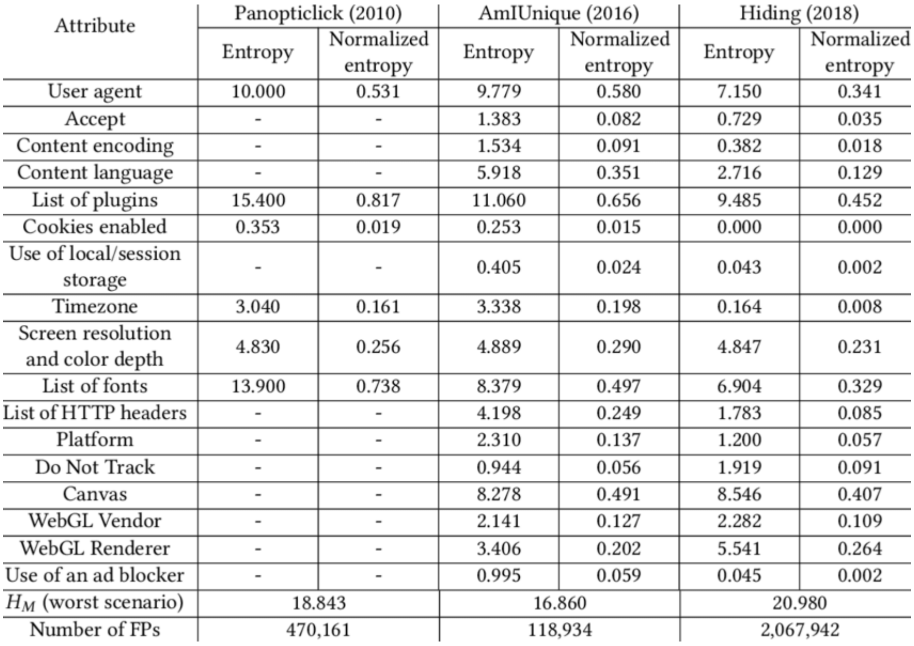
As we can see in Figure 2 provided by the Panopticlick [3], AmIUnique [9], and Hiding in the Crowd [4] studies, the biggest entropy in browser fingerprints is created by the selection of browser, the number of plugins that the user uses when searching, and also the list of fonts used by the device. A non-negligible part is also made of the language used by the user or the screen resolution of the device.

3 security risks

With fingerprinting, attackers can find weaknesses in a user's device or browser, which could result in targeted attacks. Users can be uniquely identified and tracked via their browser fingerprints, even if they use private browsing modes or get rid of their cookies. It might be challenging for users to stay anonymous online after being exposed to profiles created with browser fingerprinting. To reduce the uniqueness of their fingerprint, some users may disable or alter browser security measures [2], which then compromise their online security.

4 potential solutions

The easiest way to decrease the uniqueness of our browser fingerprint is by using multiple search engines for different tasks. Doing this makes it a lot harder for attacker to link the user's activity



Attribute	Panopticlick (2010)		AmIUnique (2016)		Hiding (2018)	
	Entropy	Normalized entropy	Entropy	Normalized entropy	Entropy	Normalized entropy
User agent	10.000	0.531	9.779	0.580	7.150	0.341
Accept	-	-	1.383	0.082	0.729	0.035
Content encoding	-	-	1.534	0.091	0.382	0.018
Content language	-	-	5.918	0.351	2.716	0.129
List of plugins	15.400	0.817	11.060	0.656	9.485	0.452
Cookies enabled	0.353	0.019	0.253	0.015	0.000	0.000
Use of local/session storage	-	-	0.405	0.024	0.043	0.002
Timezone	3.040	0.161	3.338	0.198	0.164	0.008
Screen resolution and color depth	4.830	0.256	4.889	0.290	4.847	0.231
List of fonts	13.900	0.738	8.379	0.497	6.904	0.329
List of HTTP headers	-	-	4.198	0.249	1.783	0.085
Platform	-	-	2.310	0.137	1.200	0.057
Do Not Track	-	-	0.944	0.056	1.919	0.091
Canvas	-	-	8.278	0.491	8.546	0.407
WebGL Vendor	-	-	2.141	0.127	2.282	0.109
WebGL Renderer	-	-	3.406	0.202	5.541	0.264
Use of an ad blocker	-	-	0.995	0.059	0.045	0.002
H_M (worst scenario)	18.843		16.860		20.980	
Number of FPs	470,161		118,934		2,067,942	

Figure 2: Browser attributes, their entropy and their normalized entropy [8](Laperdrix et al., 2019)

to everything they search. On the other hand, this approach decreases user experience when searching.

As the highest entropy of browser fingerprint is created when using plugins and extensions, in theory, one way of reducing uniqueness would also be by using as few plugins as possible. But as many plugins are created to increase the security of searching on the internet, ironically, this way we would again, decrease the overall searching experience, and in addition to that, we would lower the privacy protection when trying to increase it.

4.1 Randomization

Deceiving of fingerprinters could also be achieved by using random policies, as suggested by Nikiforakis, Joosen, and Livshits in PriVaricator [12]. PriVaricator operates on the principle of not allowing the linking of the same fingerprint across multiple uses. Every time a user opens a website, the website treats the user as if they are new every time. However, this randomness can affect web page rendering and may lead to the malfunction of code on certain websites.

4.2 bucketization

Bucketisation, as explained in [15]by Wang et al., is a method of reducing the uniqueness of individual fingerprints by grouping users with similar characteristics into categories called buckets. When tracking a fingerprint, attackers are able to map it to a specific bucket, rather than directly to a specific user. This type of security improvement is often combined with noise injection – the data in the bucket is further randomized or misleading information is added into buckets. Bucketization is surely a way to complicate the work of attacker, but determined attackers can still find a way around this protection layer.

4.3 slicing

Tiancheng Li, Ninghui Li, Zhang, and Molloy proposed an even more sophisticated technique to anonymize user activity called slicing. In their article [10] they propose, that „slicing partitions the data both horizontally and vertically“. Just like bucketization, the data is divided into smaller sections, and then the values are permutated, so that there will not remain clearly visible links between data sets. But with this technique, the data is not generalized and no noise is added so the preservation of the original information is much better.

5 public awareness

6 reaction to lectures

7 conclusion

References

[1] Ahmet Aksoy, Sushil Louis, and Mehmet Hadi Gunes. Operating system fingerprinting via automated network traffic analysis. In *2017 IEEE Congress on Evolutionary Computation (CEC)*, pages 2502–2509, 2017.

[2] Tomasz Bujlow, Valentín Carela-Español, Josep Solé-Pareta, and Pere Barlet-Ros. Web tracking: Mechanisms, implications, and defenses. 07 2015.

[3] Peter Eckersley. How unique is your web browser? In *Proceedings of the 10th International Conference on Privacy Enhancing Technologies*, PETS'10, page 1–18, Berlin, Heidelberg, 2010. Springer-Verlag.

[4] Alejandro Gómez-Boix, Pierre Laperdrix, and Benoit Baudry. Hiding in the crowd: An analysis of the effectiveness of browser fingerprinting at large scale. In *Proceedings of the 2018 World Wide Web Conference*, WWW '18, page 309–318, Republic and Canton of Geneva, CHE, 2018. International World Wide Web Conferences Steering Committee.

[5] Jonathan Gurary, Ye Zhu, Riccardo Bettati, and Yong Guan. *Operating System Fingerprinting*, pages 115–139. Springer New York, New York, NY, 2016.

[6] Soroush Karami, Panagiotis Ilia, Konstantinos Solomos, and Jason Polakis. Carnus: Exploring the privacy threats of browser extension fingerprinting. In *In Proceedings of the 27th Network and Distributed System Security Symposium (NDSS)*, 2020.

[7] Navpreet Kaur, Sami Azam, Krishnan Kannoorpatti, Kheng Cher Yeo, and Bharanidharan Shanmugam. Browser fingerprinting as user tracking technology. In *2017 11th International Conference on Intelligent Systems and Control (ISCO)*, pages 103–111, 2017.

[8] Pierre Laperdrix, Nataliia Bielova, Benoit Baudry, and Gildas Avoine. Browser fingerprinting: A survey. 05 2019.

[9] Pierre Laperdrix, Walter Rudametkin, and Benoit Baudry. Beauty and the Beast: Diverting modern web browsers to build unique browser fingerprints. In *37th IEEE Symposium on Security and Privacy (S&P 2016)*, San Jose, United States, May 2016.

[10] Tiancheng Li, Ninghui Li, Jian Zhang, and Ian Molloy. Slicing: A new approach for privacy preserving data publishing. *IEEE Transactions on Knowledge and Data Engineering*, 24(3):561–574, 2012.

[11] Deepali Moad, Vikas Sihag, Gaurav Choudhary, Daniel Gerbi Duguma, and Ilsun You. Fingerprint defender: Defense against browser-based user tracking. In Ilsun You, Hwankuk Kim, Taek-Young Youn, Francesco Palmieri, and Igor Kotenko, editors, *Mobile Internet Security*, pages 236–247, Singapore, 2022. Springer Nature Singapore.

[12] Nick Nikiforakis, Wouter Joosen, and Benjamin Livshits. Privaricator: Deceiving fingerprinters with little white lies. In *Proceedings of the 24th International Conference on World Wide Web*, WWW '15, page 820–830, Republic and Canton of Geneva, CHE, 2015. International World Wide Web Conferences Steering Committee.

[13] Oleksii Starov and Nick Nikiforakis. Extended tracking powers: Measuring the privacy diffusion enabled by browser extensions. In *Proceedings of the 26th International Conference on World Wide Web*, WWW '17, page 1481–1490, Republic and Canton of Geneva, CHE, 2017. International World Wide Web Conferences Steering Committee.

[14] Antoine Vastel, Pierre Laperdrix, Walter Rudametkin, and Romain Rouvoy. Fp-stalker: Tracking browser fingerprint evolutions. In *2018 IEEE Symposium on Security and Privacy (SP)*, pages 728–741, 2018.

[15] Ke Wang, Peng Wang, Ada Waichee Fu, and Raymond Chi-Wing Wong. Generalized bucketization scheme for flexible privacy settings. *Information Sciences*, 348:377–393, 2016.

[16] Ting-Fang Yen, Yinglian Xie, Fang Yu, Roger Peng Yu, and Martin Abadi. Host fingerprinting and tracking on the web: Privacy and security implications. In *NDSS*, volume 62, page 66, 2012.