BRNO UNIVERSITY OF TECHNOLOGY
FACULTY OF INFORMATION TECHNOLOGY

Excel@FIT 2025
http://excel.fit.vutbr.cz

Comparison of Privacy Preserving Tools in Web Browsers and Extensions

Bc. Vojtěch Fiala*

Abstract

The prevalent use of user tracking on the web has stimulated the development of privacy-preserving tools such as browser extensions and privacy-oriented browsers. However, comparing their effectiveness remains challenging due to the dynamic and complex nature of the web. This work focuses on evaluating and comparing tools that prevent user tracking in web browsers, specifically those tools that block web requests. Consistent conditions are ensured across tests by capturing and replaying real web traffic in a controlled environment. The proposed system utilizes directed trees to recreate request structure and identifies blocked connections, providing deeper insights into indirectly blocked requests. The results reveal significant differences in blocking effectiveness across the tested tools, with substantial discrepancies observed even when using the same extension in Chrome and Firefox.

*xfiala61@stud.fit.vut.cz, *Faculty of Information Technology, Brno University of Technology*

1. Introduction

User tracking is a prevalent problem, with trackers present on over 90 % of websites [1]. Advertisers, in particular, rely on tracking to identify who sees their ads, allowing them to serve more targeted content and increase the chance of user engagement [2].

Various privacy tools exist to combat the tracking issue, from browser extensions to specialized browsers. This work focuses on tools that block outgoing HTTP requests to known tracking domains, improving privacy and, to some extent, page load times and data usage [3].

However, the effectiveness of these tools varies greatly, and users often lack guidance on which one to choose. This work evaluates such content-blocking tools by comparing what and how many network requests they block.

Unlike current research [3, 4], which is affected by changes in website content over time, this work uses a deterministic and repeatable testing setup to ensure consistency. It models request chains as directed trees, capturing how blocking one request can prevent several others.

2. Proposed Evaluations

This work introduces two evaluation methods that consider the relations between network requests. A tree structure, previously used to model such relations [5], serves as the foundation for both evaluations.

The first evaluation assesses blocking performance by measuring how many requests are blocked directly or transitively. The recorded requests are replayed with different content-blocking tools present, and blocking results of the tested tools are logged. Transitively blocked requests are then inferred using the request trees.

The second evaluation measures anti-tracking performance using the JShelter Fingerprint Detector. The evaluation observes calls to JavaScript APIs often misused for fingerprinting and maps them to logged resources. Based on the blocking results, determined in the same way as in the first evaluation, directly and transitively blocked API calls are identified through the request trees.

3. Methodology

Both evaluations are combined into a single evaluation system that outputs multiple performance metrics.

As shown in Figure 1, the system is modular, comprising four parts that can run independently:

- **Traffic Logger** builds the dataset using a Selenium-based browser. It logs network requests, DNS responses, and fingerprinting API calls.
- **Traffic Parser** processes the logs to reconstruct request trees, where each node represents a requested resource. Each node is linked to any associated fingerprinting API calls, which are grouped (per JShelter configuration file) into *Browser Properties*, *Algorithmic Methods* and *Crawl Fp Inspector*, the latter representing APIs often used for fingerprinting [6]. Figure 2 shows an example of a request tree with associated API calls.
- **Simulation Engine** replays the observed requests. A custom web server is visited and original requests are fetched; DNS responses are controlled to provide the logged replies. Browsers are launched with various content-blocking extensions (or none if the browser itself has content-blocking capabilities) to record which requests they block. A firewall prevents unnecessary contact with original servers.
- **Analysis Engine** loads and propagates the blocking results through the request trees. The trees are then analyzed to compute several metrics that describe the behavior of the tested content-blocking tool. The results are saved and can later be used to compare multiple tools.

4. Results & Insights

Tool	Blocked Directly	Blocked Transitively	Blocked in Total
Avast Secure Browser	4,484	8,375	12,859
Brave browser	9,513	13,908	23,421
Firefox browser	147	17	164
Chrome Adblock Plus	4,132	6,011	10,143
Firefox Adblock Plus	4,132	6,011	10,143
Chrome Ghostery	9,286	13,425	22,711
Firefox Ghostery	9,133	15,596	24,729
Chrome uBlock Origin Lite	9,238	15,070	24,308
Firefox uBlock Origin	9,202	15,085	24,287

Table 1. Results of evaluation of requests blocked directly, transitively, and in total.

Tool	Browser Properties	Algorithmic Methods	Crawl Fp Inspector
Avast Secure Browser	29,556	877	635
Brave browser	55,453	1,183	1,175
Firefox browser	136	25	36
Chrome Adblock Plus	22,723	621	640
Firefox Adblock Plus	22,723	621	640
Chrome Ghostery	54,228	1,184	764
Firefox Ghostery	65,235	1,273	711
Chrome uBlock Origin Lite	64,717	1,276	708
Firefox uBlock Origin	64,972	1,276	707

Table 2. Results of evaluation of directly blocked calls to APIs potentially usable for fingerprinting.

Tool	Browser Properties	Algorithmic Methods	Crawl Fp Inspector
Avast Secure Browser	51,025	1,990	812
Brave browser	114,959	3,179	1,840
Firefox browser	7	0	0
Chrome Adblock Plus	39,832	1,597	468
Firefox Adblock Plus	39,832	1,597	468
Chrome Ghostery	113,912	3,058	1,476
Firefox Ghostery	139,537	5,010	1,490
Chrome uBlock Origin Lite	136,687	4,985	1,332
Firefox uBlock Origin	136,691	4,985	1,332

Table 3. Results of evaluation of transitively blocked calls to APIs potentially usable for fingerprinting.

Dataset used for the evaluations consists of 937 pages obtained from the *DataForSEO* list of the Top 1000 websites[1] for Czechia. A total of *100,753* requests were analyzed, along with *307,226* Browser Properties, *16,011* Algorithmic Methods, and *7,460* Crawl Fp Inspector API calls.

Figure 3 visualizes the total number of blocked requests, both direct and transitive, as detailed in Table 1. The more requests a tool blocks, the less network bandwidth should be required.

Figure 4 shows total blocked fingerprinting API calls for the *Browser Properties* group, calculated as the sum of directly and transitively blocked *Browser Properties* entries in Tables 2 and 3.

Figure 5 similarly shows the total blocked calls for *Algorithmic Methods*, combining direct and transitive results from the Tables 2 and 3.

Overall, the blocking effectiveness varies significantly, not only across tools but also between the Chrome and Firefox versions of the same extension. *Ghostery* for Firefox performed the best, closely followed by *uBlock Origin* and *uBlock Origin Lite*. Many tested tools blocked over 20 % of all network requests, underscoring the significance of using content blockers to reduce unnecessary or potentially invasive web traffic.

Acknowledgements

I would like to thank my supervisor, Ing. Libor Polčák, Ph.D., for the time he spent consulting with me, his advice, detailed feedback, and patience. I'd also like to thank Iskander Sanchez-Rola, PhD. for our consultations.

References

[1] Savino Dambra, Iskander Sanchez-Rola, Leyla Bilge, and Davide Balzarotti. When sally met trackers: Web tracking from the users' perspective. In *31st USENIX Security Symposium*

[1] https://dataforseo.com/free-seo-stats/top-1000-websites

(USENIX Security 22), pages 2189–2206, Boston, MA, August 2022. USENIX Association.

[2] Bennett Cyphers and Gennie Gebhart. Behind the one-way mirror: A deep dive into the technology of corporate surveillance. online, 2019.

[3] Ismael Castell-Uroz, Rubén Sanz-García, Josep Solé-Pareta, and Pere Barlet-Ros. Demystifying content-blockers: Measuring their impact on performance and quality of experience. *IEEE Transactions on Network and Service Management*, 19(3):3562–3573, 2022.

[4] Siddharth M. Madikeri and Vijay K. Madisetti. Ad blockers & online privacy: A comparative analysis of privacy enhancing technologies (pet). *Journal of Software Engineering and Applications*, 17:378–395, 2024.

[5] Iskander Sanchez-Rola, Matteo Dell'Amico, Davide Balzarotti, Pierre-Antoine Vervier, and Leyla Bilge. Journey to the center of the cookie ecosystem: Unraveling actors' roles and relationships. In *2021 IEEE Symposium on Security and Privacy (SP)*, pages 1990–2004, 2021.

[6] Umar Iqbal, Steven Englehardt, and Zubair Shafiq. Fingerprinting the fingerprinters: Learning to detect browser fingerprinting behaviors. In *2021 IEEE Symposium on Security and Privacy (SP)*, pages 1143–1161, 2021.