Towards Understanding Server-Side Tracking Ecosystems

Christian Böttger, Tareq Khouja

Westfälische Hochschule

if(is) internet-sicherheit.

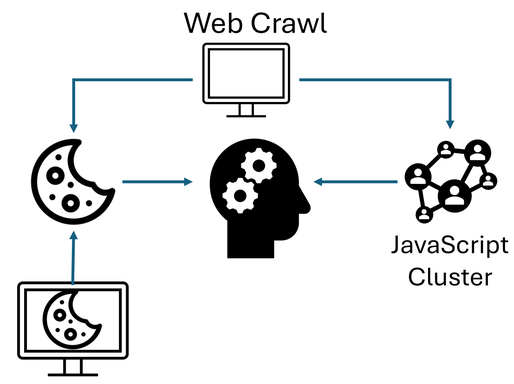
With funding from the:

Federal Ministry of Research, Technology and Space

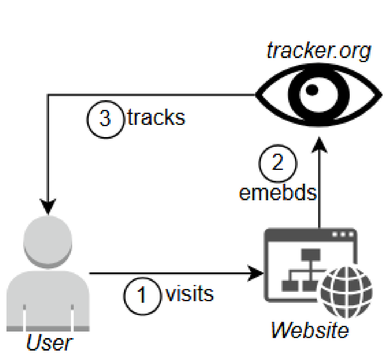
Abstract

In recent years, tracking has shifted from third-party to first-party contexts, partly driven by server-side tracking (SST). SST enables tracking without relying on third-party cookies or direct browser requests, bypassing traditional ad-blocking methods. However, this development introduces new privacy challenges, as current ad-blocking solutions fail to protect users against SST. Prior research has addressed only limited aspects of SST detection, highlighting a significant knowledge gap. This poster presents a novel approach utilizing machine learning techniques to detect server-side tracking comprehensively. Our methodology involves extensive data collection through large-scale measurements, systematically analyzing cookies and JavaScript usage from multiple parties. By identifying key indicators of SST, we try to create actionable defensive strategies, including filter-list patterns to enhance user protection. Additionally, we offer insights into the evolving historical landscape of first-party and third-party tracking practices. Our work aims to advance the understanding of server-side tracking detection and inform the development of effective countermeasures.

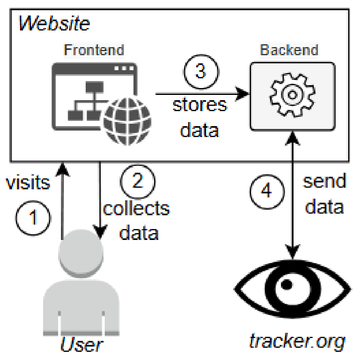
Data Processing Pipeline



Difference between Client- and Server-side Tracking



(a) Client-side tracking. The tracker is embedded into the website and directly interacts with the users.



(b) Server-side tracking. Note that there is no direct communication between the tracker and the user.

Rulebased SST Blocking

General Overview

We present a rule-based detection method to identify server-side tracking (SST) by leveraging statistical correlations in URL structures. Our approach focuses on patterns within URL paths and query parameter values, which are often indicative of tracking behavior. To obtain ground truth, we use domain-level tracking annotations from Cookiepedia and label individual URLs accordingly.

Heuristic

To construct effective detection rules, we use a correlation-based heuristic that favors patterns with the greatest difference in prevalence between tracking and non-tracking URLs. This difference, called the Correlation Gap, is defined as:

$$\text{CorrelationGap}(p) = |\Pr(p \mid \text{tracking}) - \Pr(p \mid \text{non-tracking})|$$

Patterns with a high correlation gap strongly indicate one class and rarely appear in the other, making them ideal for distinguishing tracking behavior. We rank and select patterns with the highest gaps to generate effective rules.

Evaluation Pipeline

To evaluate effectiveness, we apply the generated rules to a held-out labeled test set and report standard metrics, including true positive rate, false positive rate, and overlap with EasyList. We further conduct a page breakage test to assess the usability impact of the rules on real websites. This ensures that our defense mechanism maintains a balance between privacy protection and user experience, a crucial trade-off often overlooked in automated tracking detection.

Data Collection & Preprocessing

Besides historical data from the HTTP Archive, we measure about **25.000** domains with **15** subpages each [1]. Therefore, we use OpenWPM on four machines [2]. We used a VPN to simulate access from the US and Germany.
As input data for the heuristic we introduce later, we need two kinds of data. First, we need cookies classified by their purpose (e.g., *Target/Advertising* or *Strictly Necessary*) [3]. In addition to the name and category of the cookie, we need the URL that includes the cookie for further analysis. Second, JavaScript is needed. During the measurement, we create a SimHash over the JavaScript. Based on the SimHash, we build a similarity cluster, so that JavaScripts with a similarity of $k = 3$ are in the same cluster. Besides JavaScript, the URL of the party that includes the JavaScript is part of the data.

Shift from Third- to First-Party Tracking

General Overview

We want to analyze the shift from third-party to first-party tracking. Therefore, we used the dataset from the HTTP Archive between 2021 and 2024.

- **51,812,013** distinct cookies by name and **171,735,976** by the tuple of name, path, domain for **25,201,602** domains
- **670,237** cookies appear in first- and third-party contexts
- **488,337** where in the third-party before shifting to the first-party context

Analysis on eTLD+1 Level

We analyzed cookies based on eTLD+1 domains: (1) **17.6M** eTLD+1 domains with an average of **17 cookies each (min: 0, max: 4.8M, SD: 984)** and (2) **9.4M** distinct cookies by name. We define a cookie as $C_i = (d_i, n_i, p_i)$ (domain = site where the cookie appears; name = cookie name; party = true if first-party, false if third-party). In the following, we want to do two key analyses:

1. Cookies per domain ($COUNT(c)overeTLD$) and domains per cookie ($COUNT(eTLD)overn$)
2. Party migration, defined as a cookie with the same domain and name appearing in both first- and third-party contexts ($m = (d_i == d_{i+1}; n_i == n_{i+1}; p_i \neq p_{i+1})$).

Contact Us

Christian Böttger

Tareq Khouja

References

[1] HTTP Archive. *The HTTP Archive Tracks How the Web is Built*. https://httparchive.org/. 2025.

[2] Steven Englehardt et al. "Cookies That Give You Away: The Surveillance Implications of Web Tracking". In: *Proceedings of the 24th International Conference on World Wide Web*. WWW '15. Florence, Italy: International World Wide Web Conferences Steering Committee, 2015, pp. 289–299. ISBN: 9781450334693. DOI: 10.1145/2736277.2741679. URL: https://doi.org/10.1145/2736277.2741679.

[3] OneTrust LLC. *Cookiepedia – All You Need to Know About Cookies*. 2025. URL: https://cookiepedia.co.uk/.