Escaping the Confines of Time: Continuous Browser Extension Fingerprinting Through Ephemeral Modifications

Konstantinos Solomos
University of Illinois at Chicago
Chicago, IL, USA
ksolom6@uic.edu

Panagiotis Ilia
University of Illinois at Chicago
Chicago, IL, USA
pilia@uic.edu

Nick Nikiforakis
Stony Brook University
Stony Brook, NY, USA
nick@cs.stonybrook.edu

Jason Polakis
University of Illinois at Chicago
Chicago, IL, USA
polakis@uic.edu

ABSTRACT

Browser fingerprinting continues to proliferate across the web. Critically, popular fingerprinting libraries have started incorporating extension-fingerprinting capabilities, thus exacerbating the privacy loss they can induce. In this paper we propose *continuous fingerprinting*, a novel extension fingerprinting technique that captures a critical dimension of extensions' functionality that allowed them to elude *all* prior behavior-based techniques. Specifically, we find that *ephemeral modifications* are prevalent in the extension ecosystem, effectively rendering such extensions invisible to prior approaches that are confined to analyzing snapshots that capture a single moment in time. Accordingly, we develop Chronos, a system that captures the modifications that occur throughout an extension's life cycle, enabling it to fingerprint extensions that make transient modifications that leave no visible traces at the end of execution. Specifically, our system creates behavioral signatures that capture nodes being added to or removed from the DOM, as well as changes being made to node attributes. Our extensive experimental evaluation highlights the inherent limits of prior snapshot-based approaches, as Chronos is able to identify 11,219 unique extensions, increasing coverage by 66.9% over the state of the art. Additionally, we find that our system captures a unique modification event (i.e., *mutation*) for 94% of the extensions, while also being able to resolve 97% of the signature collisions across extensions that affect existing snapshot-based approaches. Our study more accurately captures the extent of the privacy threat presented by extension fingerprinting, which warrants more attention by privacy-oriented browser vendors that, up to this point, have focused on deploying countermeasures against other browser fingerprinting vectors.

CCS CONCEPTS

• **Security and privacy → Browser security**.

KEYWORDS

Online Tracking, Browser Fingerprinting, Extension Fingerprinting

CCS '22, November 7–11, 2022, Los Angeles, CA, USA.

ACM ISBN 978-1-4503-9450-5/22/11…$15.00
https://doi.org/10.1145/3548606.3560576

ACM Reference Format:
Konstantinos Solomos, Panagiotis Ilia, Nick Nikiforakis, and Jason Polakis. 2022. Escaping the Confines of Time: Continuous Browser Extension Fingerprinting Through Ephemeral Modifications. In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security (CCS '22), November 7–11, 2022, Los Angeles, CA, USA.* ACM, New York, NY, USA, 14 pages. https://doi.org/10.1145/3548606.3560576

1 INTRODUCTION

Modern web browsers offer an expansive collection of features and capabilities for improving the user experience, while still allowing users to further expand browsers' functionality or personalize their experience by installing browser extensions. The prevalence of extensions is made evident by a report from Google stating that "nearly half of all Chrome desktop users actively use extensions" [4]. However, this personalization suffers from inherent privacy risks: (i) the list of installed extensions can augment the browser fingerprint that websites generate for a given device, (ii) the intended functionality of extensions can reveal sensitive or personal data about the user (e.g., religion, medical issues, and nationality) [22]. In other words, extension fingerprinting presents an *additional* form of privacy loss compared to "traditional" browser fingerprinting vectors.

Nonetheless, while other browser attributes and characteristics that contribute to browser fingerprints can be trivially obtained through dedicated JavaScript APIs, no such capability exists for obtaining the list of installed extensions. Instead, the presence of a given extension needs to be inferred through implicit techniques. In fact, in recent years the research community has demonstrated various techniques for achieving that goal. Early studies relied on detecting the presence of specific web-accessible resources [42], a technique which can be rendered ineffective by countermeasures deployed by certain browsers or proposed by the research community [41, 48]. A more robust approach relies on inferring the presence of extensions based on the side-effects of their executed functionality (i.e., modifying the page's DOM [22, 46] or altering the page's stylistic properties [28]). More importantly, while extension fingerprinting has mostly been confined to academic studies,[1] recent versions of `FingerprintJS` [24] (the most prevalent browser fingerprinting library) actually incorporate such capabilities for fingerprinting extensions based on traces found in the DOM. This move has pushed extension fingerprinting into the realm of real-world privacy threats that can affect users at a wide scale.

A core limitation of *all* prior studies that infer the presence of an extension by detecting side-effects caused by its execution (i.e., DOM changes), is that they ignore their execution *life cycle* and

[1] LinkedIn being the one notable exception [36].