Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

Department of Computer Science

Exposé

Implementation of browser fingerprinting recognition and integration into PrivacyScore

Tronje Krabbe

July 30, 2017

This is an overview of the author's thesis. It describes the user identification and tracking technique 'browser fingerprinting', and explores the possibilities and technicalities in regards to detecting wether a website employs this technique.

Contents

1 Introduction

When browsing the web, users can be tracked through the use of cookies, which store information on the user's computer. If the user wishes not to be trackable, they can modify or delete any cookie as they see fit. This interaction gives operators of websites the ability to track users only if they wish to be tracked. There are, however, other methods of uniquely identifying a user and tracking them during their use of the internet, which is not as easily mitigated as a cookie.[2] On of these methods is called 'browser fingerprinting', and it will be the focus of this thesis.

Tracking a user does not simply mean recognizing whether a visitor to one's website is a new or an existing user. Identifying information can be passed on or sold to partners, advertisers, or even governments, in order to construct a rich browsing history of a user.

Browser fingerprinting, also known as device fingerprinting, works by analyzing a web browser's configuration and settings, such as installed fonts, language settings, time zone settings, installed add-ons, to name a few. These attributes are readliy available to be collected through JavaScript functions. Simply recording all function calls made by the JavaScript frontend of a website can reveal whether fingerprinting is likely to be taking place or not.[4][5]

1.0.1 Leading Question and Goals

The thesis presents the assertion that techniques to identify and track users across different websites without their knowledge or their ability to easily intervene, violates their privacy. The author will therefore attempt to implement a technique to recognize when a website is deploying browser fingerprinting, and along the way explore the techniques used to do so. The ultimate goal is to integrate the implementation into `https://privacyscore.org`, a web-service to test and rank websites according to the extent to which they respect their users' privacy.

1.0.2 Methods and Approach

Metric

If a website includes a browser fingerprinting library, or makes many calls to JavaScript functions that return data which is useful in building a fingerprint,

one can conclude: this website is very likely to actually generate, save, and possibly distribute a browser's fingerprint; simply put: it does browser fingerprinting. However, most modern websites use at least some of the attributes that can be used to construct a fingerprint for reasons other than identification. Analysing installed fonts has an obvious, legitimate use: correctly displaying text. Thus, it is important to work out a metric or rating system of some sort, which can (somewhat) reliably represent the likelihood that fingerprinting is, in fact, taking place. Gunes Acar, et. al. assert that the more fonts are being requested, the more likely a website is to be practising fingerprinting. This is based on findings of Eckersley, of the Panopticlick project.[1] Gunes Acar, et. al. further state that they classify a JavaScript file as a fingerprinter "when it loads more than 30 fonts, enumerates plugins or mimeTypes, detects screen and navigator properties, and sends the collected data back to a remote server."[1]

Scope

Finding or creating a single website which collects a fingerprint, and basing development of fingerprinting recognition on it, will not yield reliable or representative results. It is imperative to test the created recognition software against a multitude of fingerprinting libraries and users of these. It might be prudent to test the software against, say, the top 500 website as ranked by Alexa [1], as there is a high likelihood these sites employ a multitude of techniques to track their users, but there is no way to be sure if some of these sites wouldn't generate false positives. The safest way could be to create websites which employ an array of fingerprinting libraries both preexisting and implemented by the author, and then comparing the employed JavaScript calls with those from popular websites.

False positives

Filtering false positives is a non-trivial problem in this context. Requesting fonts via JavaScript might be a dead giveaway that fingerprinting code is, in fact, fingerprinting code. However, it might cause legitimate code, such as a multimedia player of sorts, to be mistaken for fingerprinting code, as well, even though it has a legitimate reason and perhaps a necessity to know a system's installed fonts. Meta-data can be used to filter these; a list can be compiled of popular JavaScript libraries with legitimate, yet fingerprinting-like behavior, and then extended through analysis of collected data (see Technology below).

[1] `http://www.alexa.com/topsites`

2 Implementation

TODO: Something about PrivacyScore

2.0.1 Technology

OpenWPM

"OpenWPM is a web privacy measurement framework which makes it easy to collect data for privacy studies on a scale of thousands to millions of site"[1]. The author is planning to build upon OpenWPM[3] to create a software that can detect browser fingerprinting. OpenWPM includes capabilities to record "all method calls (with arguments) and property accesses for APIs of potential fingerprinting interest", providing a good basis for data collection. Analysing and interpreting this data in a meaningful way will need to be implemented by the author.

SQL

Since OpenWPM produces Sqlite databases containing the results of its tests, a database will be used. It may prove useful to import the resulting Sqlite databases into a more sophisticated system, such as PostgreSQL, perhaps merging them in a meaningful way, and analyzing data not from a single website, but many, as one.

[1] `https://github.com/citp/OpenWPM`

Bibliography

[1] Gunes Acar et al. "FPDetective: dusting the web for fingerprinters". In: *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security*. ACM. 2013, pp. 1129–1140.

[2] *Am I Unique?* URL: `https://amiunique.org/` (visited on July 4, 2017).

[3] Steven Englehardt and Arvind Narayanan. "Online tracking: A 1-million-site measurement and analysis". In: *Proceedings of ACM CCS 2016*. 2016.

[4] Amin Faiz Khademi. "Browser Fingerprinting: Analysis, Detection, and Prevention at Runtime". PhD thesis. 2014.

[5] Electronic Frontier Foundation. *Panopticlick*. URL: `https://panopticlick.eff.org/` (visited on July 4, 2017).