Computers & Security 114 (2022) 102575

Contents lists available at ScienceDirect

Computers & Security

journal homepage: www.elsevier.com/locate/cose

TC 11 Briefing Papers

A Two-Step TLS-Based Browser fingerprinting approach using combinatorial sequences☆,☆☆

Bernhard Garn[a], Stefan Zauner[a], Dimitris E. Simos[a,*], Manuel Leithner[a], Richard Kuhn[b], Raghu Kacker[b]

[a] SBA Research, Floragasse 7, 1040 Vienna, Austria
[b] NIST, 100 Bureau Drive, Gaithersburg, MD 20899, USA

ARTICLE INFO

Article history:
Received 16 July 2021
Revised 17 November 2021
Accepted 7 December 2021
Available online 28 December 2021

Keywords:
Browser fingerprinting
TLS Protocol
Combinatorial sequences
Anonymity set
Fraud detection

ABSTRACT

We propose a two-step TLS-based fingerprinting approach using combinatorial sequences and properties of TLS handshake messages. Our approach combines fingerprinting based on attributes of the initial ClientHello message with the observed behavior of TLS clients when presented with permuted handshake messages in order to enhance the granularity of the derived fingerprints without increasing the required number of exchanged messages. We conduct a detailed evaluation against 21 browsers and TLS clients on two operating systems. The results show a significant increase in the entropy of the achieved splittings, allowing for a more precise identification of the TLS client than permitted by either of the underlying approaches in isolation.

1. Introduction

The term *browser fingerprinting* describes the process and corresponding methods for collecting data about a user's browser and system (Laperdrix et al., 2020), often with the goal of uniquely identifying a browser. In 2010, the results of one of the first large-scale empirical studies on browser fingerprinting reported in Eckersley (2010) pointed towards serious resulting privacy issues. Ensuing research has uncovered a variety of different ways in which browsers – voluntarily or involuntarily – may leak information (Laperdrix et al., 2016).

Fingerprinting finds applications in multiple domains; for example, it can be used to augment the authentication processes of web services (Alaca and Van Oorschot, 2016; Andriamilanto et al., 2021; 2020) or detect likely cases of fraud. More controversial use cases include covert tracking for the purposes of targeted advertising, where its main advantage compared to traditional methods is the absence of identifiers that must be stored in the user's browser (e.g. through the use of cookies).

While early works already identified a plethora of features that can be used to generate a browser fingerprint(Eckersley, 2010), modern web browsers have further expanded this surface by offering additional Application Programming Interfaces (APIs) that expose details of the underlying operating system and hardware. This is especially true for browsers running on mobile devices, where sensors that are typically not found on desktop hardware have been shown to be usable for fingerprinting (e.g., the audio API (Queiroz and Feitosa, 2019), the accelerometer (Bojinov et al., 2014) or the battery API (Olejnik et al., 2017)).

Most of these approaches share a common weakness: They require the use of JavaScript, which can be limited or disabled completely via browser add-ons such as NoScript (Maone, 2012). Similarly, fingerprinting methods that build upon external plugins such as Flash or Java can generally be defeated by disabling these components.

In contrast, the approach presented in this work does not rely on any external components, but merely requires the use of TLS (used in conjunction with HTTP as HTTPS (Rescorla et al., 2000)), a protocol that is widely deployed on the internet today (Felt et al., 2017) and enabled in all modern browsers.

The general domain of browser fingerprinting can be approached from two different and complementary sides: performing empirical observation-based studies via the collection of browser fingerprints in the real world or evaluating fingerprinting ap-

☆ The research presented in this paper was carried out in the context of the Austrian COMET K1 program and partly publicly funded by the Austrian Research Promotion Agency (FFG) and the Vienna Business Agency (WAW)

☆☆ Moreover, this work was performed partly under the following financial assistance award 70NANB18H207 from U.S. Department of Commerce, National Institute of Standards and Technology

* Corresponding author.
E-mail addresses: BGarn@sba-research.org (B. Garn), SZauner@sba-research.org (S. Zauner), DSimos@sba-research.org (D.E. Simos), MLeithner@sba-research.org (M. Leithner), d.kuhn@nist.gov (R. Kuhn), raghu.kacker@nist.gov (R. Kacker).

https://doi.org/10.1016/j.cose.2021.102575
0167-4048/