HAL
open science

BrFAST: a Tool to Select Browser Fingerprinting Attributes for Web Authentication According to a Usability-Security Trade-off

Nampoina Andriamilanto, Tristan Allard

To cite this version:

Nampoina Andriamilanto, Tristan Allard. BrFAST: a Tool to Select Browser Fingerprinting Attributes for Web Authentication According to a Usability-Security Trade-off. Companion Proceedings of the Web Conference 2021 (WWW '21 Companion), Apr 2021, Ljubljana, Slovenia. 10.1145/3442442.3458610 . hal-03202788

HAL Id: hal-03202788

https://hal.science/hal-03202788v1

Submitted on 20 Apr 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

BrFAST: a Tool to Select Browser Fingerprinting Attributes for Web Authentication According to a Usability-Security Trade-off

Nampoina Andriamilanto
tompoariniaina.andriamilanto@irisa.fr
Univ Rennes, CNRS, IRISA
Rennes, France

Tristan Allard
tristan.allard@irisa.fr
Univ Rennes, CNRS, IRISA
Rennes, France

ABSTRACT

In this demonstration, we put ourselves in the place of a website manager who seeks to use browser fingerprinting for web authentication. The first step is to choose the attributes to implement among the hundreds that are available. To do so, we developed BrFAST, an attribute selection platform that includes FPSelect, an algorithm that rigorously selects the attributes according to a trade-off between security and usability. BrFAST is configured with a set of parameters for which we provide values for BrFAST to be usable as is. We notably include the resources to use two publicly available browser fingerprint datasets. BrFAST can be extended to use other parameters: other attribute selection methods, other measures of security and usability, or other fingerprint datasets. BrFAST helps visualize the exploration of the possibilities during the search of the best attribute set to use, evaluate the properties of attribute sets, and compare several attribute selection methods. During the demonstration, we compare the attribute sets selected by FPSelect with those selected by the usual methods according to the properties of the resulting browser fingerprints (e.g., their usability, their unicity).

CCS CONCEPTS

• **Security and privacy** → **Multi-factor authentication**; • **Information systems** → *Browsers*.

KEYWORDS

browser fingerprinting, web authentication

ACM Reference Format:
Nampoina Andriamilanto and Tristan Allard. 2021. BrFAST: a Tool to Select Browser Fingerprinting Attributes for Web Authentication According to a Usability-Security Trade-off. In *Companion Proceedings of the Web Conference 2021 (WWW '21 Companion), April 19–23, 2021, Ljubljana, Slovenia.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3442442.3458610

1 INTRODUCTION

Browser fingerprinting [1, 4, 7, 11] is the collection of attributes from a web browser to build a potentially unique fingerprint. Initially used to track users on the web, this technique can also supplement passwords as an additional web authentication factor as depicted in Figure 1. Hundreds of attributes are available but collecting all of them is unrealistic as their usability cost (e.g., their collection time) would be too high [2]. Moreover, the attributes can be correlated with each other as depicted in Table 1. Previous studies consider a small set of usual attributes [4, 7, 11], iteratively pick the attribute of the highest entropy – or conditional entropy – until reaching a threshold [3, 5, 8, 10, 12, 14], or evaluate every possibility [6]. However, the entropy does not consider the correlations that occur between the attributes. Moreover, the entropy and the conditional entropy do not capture the usability cost induced by the use of the attributes [1]. As for the evaluation of every possibility, we emphasize that it is impractical as the number of possibilities grows exponentially with the number of attributes. We propose a demonstration of FPSelect [1], a rigorous approach to select a subset of the candidate attributes such that the cost of using the fingerprints is low and a minimum security level against dictionary attacks is reached. FPSelect helps to protect against strong dictionary attackers who have the knowledge of the fingerprint distribution among the protected users. To do so, it explores the space of the possible attribute sets using a greedy algorithm inspired by the Beam Search algorithm [9]. This demonstration illustrates how FPSelect can be used by a website manager – the verifier – who seeks to use browser fingerprinting as an additional web authentication factor. We compare the attribute sets selected by FPSelect with those selected by the usual attribute selection methods according to the properties of the resulting browser fingerprints (e.g., their usability cost, their unicity). For this end, we developed BrFAST[1], an attribute selection tool that performs the attribute selection given a set of parameters (e.g., fingerprint dataset, selection method). Users can use the set of parameters that are provided with BrFAST to perform the attribute selection, or choose their own set of parameters. We notably provide the resources to use two publicly available browser fingerprint datasets. BrFAST can be extended to use other

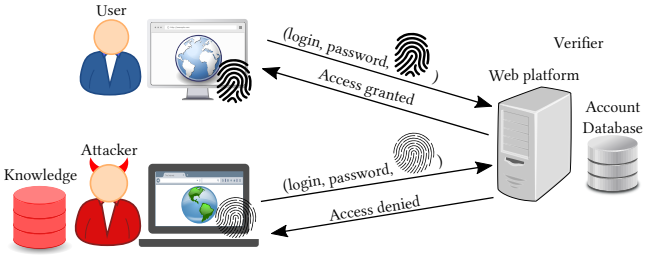

Figure 1: Example of a browser fingerprinting web authentication mechanism and a failed attack.

[1] https://github.com/tandriamil/BrFAST

This paper is published under the Creative Commons Attribution 4.0 International (CC-BY 4.0) license. Authors reserve their rights to disseminate the work on their personal and corporate Web sites with the appropriate attribution.
WWW '21 Companion, April 19–23, 2021, Ljubljana, Slovenia
© 2021 IW3C2 (International World Wide Web Conference Committee), published under Creative Commons CC-BY 4.0 License.
ACM ISBN 978-1-4503-8313-4/21/04.
https://doi.org/10.1145/3442442.3458610

User	CookieEnabled	Language	Timezone	Screen
u_1	True	fr	-1	1080
u_2	True	en	-1	1920
u_3	True	it	1	1080
u_4	True	sp	0	1920
u_5	True	en	-1	1080
u_6	True	fr	-1	1920

Table 1: Example of browser fingerprints shared by users. The `CookieEnabled` attribute provides no distinctiveness but increases the usability cost. The `Timezone` and the `Language` attributes are the two most distinctive attributes, but considering them both does not improve the distinctiveness compared to considering `Language` alone due to their correlation.

parameters: other attribute selection methods, other measures of security and usability, or other fingerprint datasets. BrFAST helps visualize the exploration of the possibilities during the search of the best attribute sets to use, evaluate the properties of attribute sets, and compare several attribute selection methods.

2 FPSELECT ALGORITHM

In this demonstration we showcase FPSelect [1], a framework to help verifiers select the browser fingerprinting attributes to design their probe. To do so, FPSelect performs a trade-off between the security that the attributes provide against a dictionary attacker and the usability cost that they induce.

2.1 Dictionary Attack and Sensitivity Measure

We consider the attackers that managed to obtain the knowledge of a fingerprint distribution (e.g., from a stolen browser fingerprint dataset). These attackers are able to submit a limited number of the most common fingerprints to impersonate as many users as possible. Given an attribute set, we measure the reach of the attackers by the proportion of the protected users that they manage to impersonate, and call this proportion the *sensitivity*. Any sensitivity measure can be plugged in FPSelect as long as it is monotonously decreasing when the number of selected attributes increases [1]. Indeed, adding an attribute should decrease the sensitivity if the attribute helps distinguish different browsers, or otherwise keep the sensitivity equal.

2.2 Usability Cost Measure

FPSelect also takes a usability cost measure as a parameter, which evaluates the usability cost of an attribute set. Any usability cost measure can be plugged in FPSelect as long as it is strictly increasing with the number of selected attributes. Indeed, adding an attribute requires at least to implement its collection, store its information, and collect it from the browser.

2.3 Lattice Model and Exploration Algorithm

FPSelect models the possibility space as a lattice of attribute sets. The elements of this lattice are the subsets of the candidate attributes and the order is the subset relationship. FPSelect leverages

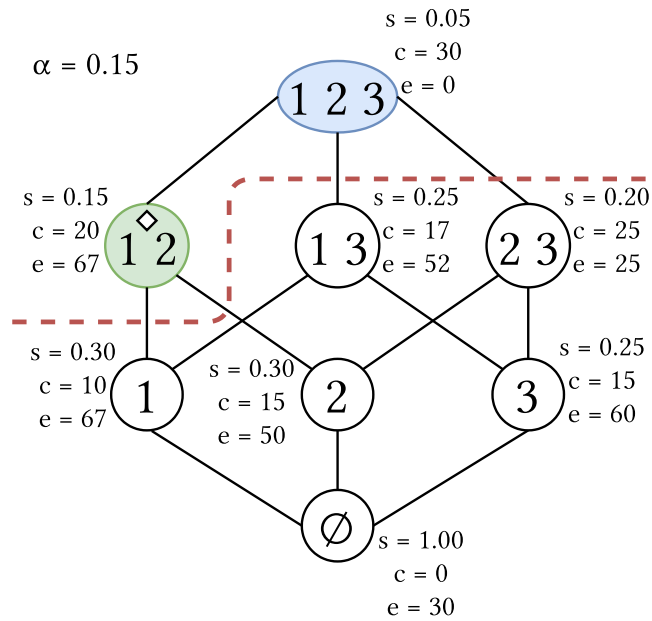

Figure 2: Example of a lattice of attribute sets, with their usability cost c, their sensitivity s, and their efficiency e. The sensitivity threshold is of $\alpha = 0.15$. The blue node satisfies the sensitivity threshold, the white nodes do not, and the green node with a diamond satisfies the sensitivity and minimizes the usability cost. The red line is the satisfiability frontier.

an exploration algorithm [1] to find the attribute set that satisfies the sensitivity threshold at a low cost. It starts from the empty set and explores k-paths in the lattice until all the paths reach the satisfiability frontier, k being a parameter. The satisfiability frontier separates the attribute sets that satisfy the sensitivity threshold from those that do not. The attribute sets right above this frontier satisfy the sensitivity threshold at a lower usability cost than their supersets. Both the optimal solution and the solution found by FPSelect are among these attribute sets. The exploration algorithm explores in priority the supersets of the most efficient[2] attribute sets and includes three pruning methods [1] to reduce the number of explored attribute sets. The exploration algorithm is inspired by the Beam Search algorithm [9] and is part of the Forward Selection algorithms [13]. The computational complexity of the exploration algorithm is of $O(kn^2\omega)$ with n being the number of candidate attributes and ω being the computational complexity of the sensitivity and usability cost measures. The memory complexity of the exploration algorithm is of $O(kn^2)$. Figure 2 shows an example of a lattice obtained from the possible attribute sets generated from three candidate attributes.

[2] The efficiency of an attribute set is the ratio between the usability cost reduction and the sensitivity. The usability cost reduction is computed as $\text{cost}(A) - \text{cost}(C)$ with C the evaluated attribute set and A the candidate attributes.

2.4 Experimental Results

We evaluated the performances of FPSelect and compared them to the baselines based on entropy and conditional entropy [1]. The experimental setting was composed of a user population of $30,000$ browsers, a number of explored paths of 1 and 3, a sensitivity threshold between 0.001 and 0.025, and a number of submissions by the dictionary attacker between 1 and 16. The sensitivity was measured as the proportion of impersonated users by the most common fingerprints, considering distance functions between attributes to allow small changes. The usability cost was measured as the weighted sum between the average fingerprint size, the average fingerprint collection time, and the proportion of attribute changes among the observed consecutive fingerprints. Compared to the attribute sets found by the baselines, those found by FPSelect generate fingerprints having a size 12 to $1,663$ times lower, a collection time 9 to $32,330$ times lower, and 4 to 30 times less attribute changes between the consecutive fingerprints. Although FPSelect explores three orders of magnitude more attribute sets compared to the baselines, the usability cost reduction is reflected on each authentication performed by each user.

3 ATTRIBUTE SELECTION TOOL

We have implemented FPSelect and wrapped it into a full-fledged attribute selection tool: BrFAST. BrFAST is configured with a set of parameters used to process the attribute selection, for which we provide values for anyone to directly use BrFAST as is. BrFAST is modular: other attribute selection methods or measure functions can be plugged-in easily. As the attribute selection process can take time, BrFAST supports the replay of execution traces. We developed BrFAST as a web application in Python3, used Flask[3] for the web application, and used D3.js[4] for the visualization of the lattice exploration.

3.1 Parameters of the Attribute Selection Tool

An attribute selection method. The implemented attribute selection methods are the entropy and the conditional entropy, together with FPSelect which is configured with the number of paths explored in the lattice of the possibilities.

A browser fingerprint dataset. The fingerprint dataset is collected from the browser population to protect with the fingerprints being composed of the complete set of attributes. BrFAST includes the resources needed to use two publicly available browser fingerprint datasets. The first dataset[5] is a sample of the dataset used in the FPStalker study [15] and the second comes from an experimentation processed by Henning Tillmann[6].

Sensitivity and usability cost measures. BrFAST includes a sensitivity and a usability cost measure inspired by [1] that can be trained on the two provided fingerprint datasets. The sensitivity is measured by the proportion of the users that share the k most common fingerprints, with k a parameter set by the verifier. The usability captures the memory size and the instability of the generated fingerprints.

A sensitivity threshold. The sensitivity threshold is configured by the verifier according to her security requirements.

3.2 Visualizations

BrFAST helps understand the inner working of FPSelect, visualize the properties of the selected attributes, and compare the attribute selection methods. The inner working of FPSelect is visualized by the real-time exploration of the lattice of the possibilities – similar to Figure 2 – and the best solution currently found. The properties of an attribute set include its usability cost, its sensitivity, a sample of the resulting fingerprints together with their entropy, their unicity, and their stability. Using the visualization of the properties of the selected attributes, BrFAST helps to compare several attribute selection methods.

4 SCENARIO

In this demonstration, we showcase FPSelect by comparing its results with those of the baselines using BrFAST. As the attribute selection process can take time, we will replay traces of executions on fingerprint datasets and sets of parameters. These traces will be available for the audience to replay them. Moreover, the audience can also plug fingerprint datasets, sensitivity and usability cost measures, and sets of parameters.

5 CONCLUSION

In this demonstration, we put ourselves in the place of a website manager that seeks to use browser fingerprinting as an additional web authentication factor. To do so, she has to choose the attributes to collect to compose the browser fingerprints. For this purpose, we developed BrFAST, an attribute selection tool that embarks the FPSelect algorithm to rigorously select browser fingerprinting attributes according to a trade-off between security and usability. Using BrFAST, we compare the attribute sets that are found by FPSelect and by the usual attribute selection methods, as well as the resulting browser fingerprints.

REFERENCES

[1] Nampoina Andriamilanto, Tristan Allard, and Gaëtan Le Guelvouit. 2020. FPSelect: Low-Cost Browser Fingerprints for Mitigating Dictionary Attacks against Web Authentication Mechanisms. In *Annual Computer Security Applications Conference (ACSAC)* (2020). https://doi.org/10.1145/3427228.3427297

[2] Nampoina Andriamilanto, Tristan Allard, Gaëtan Le Guelvouit, and Alexandre Garel. 2020. A Large-scale Empirical Analysis of Browser Fingerprints Properties for Web Authentication. (2020). https://arxiv.org/abs/2006.09511

[3] C. Blakemore, J. Redol, and M. Correia. 2016. Fingerprinting for Web Applications: From Devices to Related Groups. In *IEEE Trustcom/BigDataSE/ISPA* (2016-08). 144–151. https://doi.org/10.1109/TrustCom.2016.0057

[4] Peter Eckersley. 2010. How Unique is Your Web Browser?. In *International Conference on Privacy Enhancing Technologies (PETS)* (2010). 1–18. https://doi.org/10.1007/978-3-642-14527-8_1

[5] David Fifield and Serge Egelman. 2015. Fingerprinting Web Users Through Font Metrics. In *Financial Cryptography and Data Security (FC)* (2015), Rainer Böhme and Tatsuaki Okamoto (Eds.). 107–124. https://doi.org/10.1007/978-3-662-47854-7_7

[6] Erik Flood and Joel Karlsson. 2012. Browser Fingerprinting. https://hdl.handle.net/20.500.12380/163728

[7] Alejandro Gómez-Boix, Pierre Laperdrix, and Benoit Baudry. 2018. Hiding in the Crowd: an Analysis of the Effectiveness of Browser Fingerprinting at Large Scale.

[3] https://flask.palletsprojects.com
[4] https://d3js.org
[5] https://github.com/Spirals-Team/FPStalker
[6] https://www.henning-tillmann.de/2013/10/browser-fingerprinting-93-der-nutzer-hinterlassen-eindeutige-spuren

In *The Web Conference (TheWebConf)* (2018-04). https://doi.org/10.1145/3178876.3186097

[8] Peter Hraška. 2018. Browser Fingerprinting. https://virpo.sk/browser-fingerprinting-hraska-diploma-thesis.pdf

[9] Daniel Jurafsky and James H. Martin. 2009. *Speech and Language Processing (2Nd Edition).* Prentice-Hall, Inc., Upper Saddle River, NJ, USA. 325–326 pages.

[10] Amin Faiz Khademi, Mohammad Zulkernine, and Komminist Weldemariam. 2015. An Empirical Evaluation of Web-Based Fingerprinting. 32, 4 (2015), 46–52. https://doi.org/10.1109/MS.2015.77

[11] Pierre Laperdrix, Walter Rudametkin, and Benoit Baudry. 2016. Beauty and the Beast: Diverting Modern Web Browsers to Build Unique Browser Fingerprints. In *IEEE Symposium on Security and Privacy (S&P)* (2016-05). 878–894. https://doi.org/10.1109/SP.2016.57

[12] João Pedro Figueiredo Correia Rijo Mendes. 2011. noPhish – Anti-phishing System using Browser Fingerprinting. https://estagios.dei.uc.pt/cursos/mei/relatorios-de-estagio/?id=279

[13] Rachel Schutt and Cathy O'Neil. 2014. *Doing data science: Straight talk from the frontline.* O'Reilly. 181–182 pages.

[14] Kazuhisa Tanabe, Ryohei Hosoya, and Takamichi Saito. 2018. Combining Features in Browser Fingerprinting. In *Advances on Broadband and Wireless Computing, Communication and Applications (BWCCA)* (2018), Leonard Barolli, Fang-Yie Leu, Tomoya Enokido, and Hsing-Chung Chen (Eds.). 671–681. https://doi.org/10.1007/978-3-030-02613-4_60

[15] Antoine Vastel, Pierre Laperdrix, Walter Rudametkin, and Romain Rouvoy. 2018. FP-STALKER: Tracking Browser Fingerprint Evolutions. In *IEEE Symposium on Security and Privacy (S&P)* (2018-05-21). 728–741. https://doi.org/10.1109/sp.2018.00008