Browser Fingerprint Standardization Using Rule-Based Algorithm And Multi-Class Entropy

Elbren Antonio, Arnel Fajardo, Ruji Medina

Abstract: Over the years, internet users believed that IP addresses and Cookies are the only software application tool for digital fingerprints to track people online. Modern web technologies allowed interested organizations to use new ways to identify and track users without their knowledge and provide no way to avoid it. Browser fingerprints or device fingerprint replaces the concept of IP addresses and Cookies. Our works explores how reliable this collected information from browser to manage the stability and diversity of devices information due to continuously software upgrades with or without the user's knowledge. The researcher focuses on identifying returning devices online using the collected dataset based on 16 device information attributes. Finally, based on our findings, we discuss the current understanding of fingerprinting by identifying devices instead of identifying the user.

Index Terms: Browser Fingerprint, Diversity, Entropy, Online Privacy, Rule Based Algorithm, Stability, Web Browser

1 INTRODUCTION

Web browsers are considered the most important software application in accessing information on the world. Internet users navigate through pages and manage this tool depending on the user's preferences. Using this tool, users did not realize that they are tracked by advertisers, social media widgets, and web- site analytics engines [1],[2]. Second generation tracking method called browser fingerprinting occurs, moving from stateful identifiers to stateless [3], and that means, there is no recording of preceding connections and each communication request has to be handled based on collected information that comes with it. Yet, device fingerprints distinctiveness, by itself, is not enough for tracking because fingerprints change. To keep track of these fingerprint growth and associate to previous fingerprints. Recent approaches exploit fingerprint uniqueness as a defense mechanism by adding randomness to break uniqueness [4],[5],[6], but tracking a device is not given attention to prove how accurate it is in the wild. Third-party web tracking [7], in practice in which the tracker is another entity other than the website directly visited by the user, goes beyond monitoring the user's visit to the site [8], while combined tracking data is often related with a unique advertising identifier, it can be counter instinctive when the tracker happens to be a third-party. Further studies have focused on studying new attributes that increase browser fingerprint uniqueness [9],[10], while others have exposed that websites use browser fingerprinting as a way to regenerate deleted cookies [11]. This research provides an analysis using some rules identify device through browser information. The results relies on browser information through http headers, JavaScript and or flash detection [12], which is collected through the orangeecko.net website. The device information through browser includes 16 attributes for evaluation. We access some of these attributes by means of to the most recent web technologies, such as, the HTML5 canvas element as considered the most promising attribute in fingerprinting [13].

TABLE 1

BROWSER FINGERPRINT EXAMPLE SHOWING THE ATTRIBUTES TRIGGERS WITH EXAMPLE VALUE.

Attribute	Triggers	Value
browser	Automatic	Chrome 74
flash	User	N/A
canvas	Automatic	…
connection	Automatic	cellular
cookie	User	true
display	Context	24|1536|864|1536|824
font smoothing	Automatic	Unknown
fonts	Automatic	Agency FB|Arial|Bell MT|Bodoni MT|Calibri
form fields	Automatic	url=http://orangeecko.net/info.php
java	Automatic/User	false
language	User	lang=en-US|syslang=|userlang=
silverlight	Automatic/User	N/A
os	Automatic	Windows NT 4.0|32 bits
timezone	Context	-7
touch	Automatic	false
true browser	Automatic	Safari
plugins	Automatic/User	N/A
user agent	Automatic	mozilla/5.0 (windows nt 10.0; win64; x64) applewebkit/537.36 (khtml, like gecko)

Table shows how attributes values are pulled based on triggers. Some values specifically drive by automatic changes without user interactions.

1.1 TRIGGERS

Browser identifications evolves in the following reasons and categorized by the following type of triggers:

1. **Automatic.** This is happening automatically and without direct user consent. User agent in this case is affected by this software updates and or software upgrades [14];

- *Elbren Antonio, Technological Institute of the Philippines, Manila, Philippines. E-mail: breeve.antonio@gmail.com*
- *Arnel Fajardo, Manuel L. Quezon University, School of Engineering and Information Technology, Manila, Philippines. E-mail: acfajardo2011@gmail.com*
- *Ruji Medina, Technological Institute of the Philippines, Manila, Philippines. E-mail: medina@tip.edu.ph*

2. **Context-dependent.** users' settings and concept and or how the user configure their personal devices like monitor resolutions [15], internet speed and because of location changes, this is indirectly obstructed by a contextual; and

3. **User-triggered.** It requires user intervention on specific attributes specially by allowing cookies or local storage in the browser.

1.2 BROWSER FINGERPRINT STABILITY AND DIVERSITY

Stability and diversity of device information growths by means of fingerprinting as a lasting technique to track web users requires not only obtaining unique browser fingerprints. Most of the works and previous studies has focused on increasing fingerprint uniqueness [16]. Uniqueness is a property of fingerprints that is critical to evaluate due to its small amount of information, it is also critical to understand fingerprint evolution to build an effective tracking technique. This study provides more insights into browser fingerprint evolution in order to demonstrate the effectiveness of such a tracking technique.

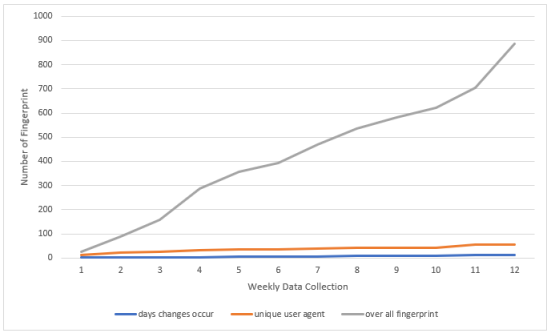

Fig. 1. Number of distinct user agent occurrences per week caused by browser upgrades/downgrades.

1.3 DATASET

Collected datasets contains less than a thousand device information from different browser instances. Browser fingerprints are obtained from identified computer laboratories and known mobile, laptops and other devices from March 2019 up to the present and continue growing from invited and willing participants using their personal devices information. All participants are invited through personal communication and social media to gather enough dataset. The website www.orangeecko.net is setup to gather data, we manage to add *ip address* attribute to track visitors all the time. Another website, www.orangeecko.net/tutorials is also launch to collect new dataset and setup by adding identification of each user and login from different devices using their user's identifications.

In summary, this research focuses on the following contributions:

1. Identify returning devices using device information.
2. Implement modified rule-based algorithm to track device information.
3. Identify how promising the selected attributes in identifying a device.

2 LITERATURE REVIEW

Device tracking is becoming persistent as advertisers and tracking companies seek to refine their targeting, detect fraud, or offer new services. While most of today's tracking is done through third-party cookies, prior research has shown that browser and system attributes can be used to uniquely identify devices through fingerprints [17]. Web-based device fingerprinting is the process of collecting security information through the browser to perform stateless device identification and become the source of privacy problem because of their being uniqueness and stability [18]. Fingerprints may then be used to identify and track computing devices in the web. Browser fingerprinting has emerged as a technique to track users without their permission. Contrasting to cookies, fingerprinting is a stateless [19] technique that does not store any information on client devices, instead it exploits unique attributes provided by users' browser. The uniqueness of fingerprints allows them to be used for identification [20]. Electronic Frontier Foundation (EFF) were the first to explore the degree of web browsers for "device fingerprinting" via the version and configuration information of the devices collected that they will transmit to websites upon request. Panopticlick also measures the distinctiveness of browser instances by anonymously log the specific information, and compare it to a database of many other data they collect [21]. AmIUnique.org implemented a browser fingerprinting script that exploits state-of-the-art techniques as well as some new browser APIs. AmIUnique work provides an in-depth analysis of the extent to which today's web provides an effective means to uniquely identify users through browser fingerprinting. This analysis relies on more than 118,000 fingerprints, which is collected through the AmIUnique.org website.Uniqueness of DOM element is also discovered that almost 90% is uniquely recognizable by combination of change and at least 86% is one of them completely distinct [12],[22].Unique information provided by browsers are generated whenever users visit a website by collecting information without the knowledge of individual visitors. This information is added to the other information gathered to the same website to provide unique fingerprints. Users has no access on this data to manipulate and control while it is generated from the server [23],[24]. HTML 5 canvas fingerprinting also bring much attention because of its consistency no matter how several times this canvas to be rendered it returns the same canvas unique identification [25].

3 METHODOLOGY

Due to the large number of possible features that can serve as fingerprints, this study focuses only on the selected fingerprint features, which it seems to be the common attributes found in any other research that can identify returning devices on the same website.

TABLE 2
BROWSER FINGERPRINT MEASUREMENT DATA EXAMPLE

Attribute	Value	Unique	Distinct	Entropy
Browser	Chrome 79	15	0	2.4408371252877
Canvas	1855019831	31	0	2.9034929700652
Display	24|1680|1050|1680|1010	28	2	2.2625909730343
Fonts	Agency FB|Arial Black|Bodoni MT|Calibri Light|Castellar|Colonna MT|Consolas|Constantia|Copperplate G...	33	2	2.7510212800625
Language	lang=en-US|syslang=|userlang=	4	923	0.23310850297216
OS	Windows NT 4.0|32 bits	8	39	0.90382565723649
User Agent	mozilla/5.0 (windows nt 10.0; win64; x64) applewebkit/537.36 (khtml, like gecko) chrome/79.0.3945.130 safari/537.36|Win32|en-US	73	0	3.7656640958281
Time Zone	8	4	119	0.61772292018068
True Browser	Safari	2	522	0.99359738932612
Plug-Ins	chrome pdf plugin|chrome pdf viewer	21	4	2.5410718472459
Font Smoothing	Unknown	2	519	0.99442368206282
Java	False	2	933	0.15803643298196
Connection	undefined	4	444	1.3056794928016
Touch	false	2	899	0.32177167437138
Form Fields	url=http://orangeecko.net/tutorials/verify.php	2	0	0.18159890418458
Silverlight	N/A	2	946	0.076884331138132

3.1 Mathematical Treatment

We measure the the collected information using multi-class entropy. The formula for entropy generalizes to more classes, the general formula:

$$Entropy = -\sum_{i=1}^{n} pi\ log_2\ pi \qquad (1)$$

The general formula for multi-class entropy where there are *n* classes, and *p_i* is the probability an object from the *i-th* class appearing in the dataset. In the case of our dataset, the 16 attributes are grouped by classes.

3.2 Rule-based Algorithm

In this study, the following rules are applied to identify the returning user's device:

1. New device information is not kept or added to database instead it is compared to all previous collected dataset from different devices.
2. Evaluate most promising attributes like *browser, canvas, display, fonts* and *user-agent* as the first basis of identifying devices.
3. Calculate the similarity of each attribute using multi-class entropy and get the sum of each attributes to provide fingerprints results.

3.3 Results

To test the dataset, we first collect device information from selected participant to demonstrate the changes of each device information within 74 days.

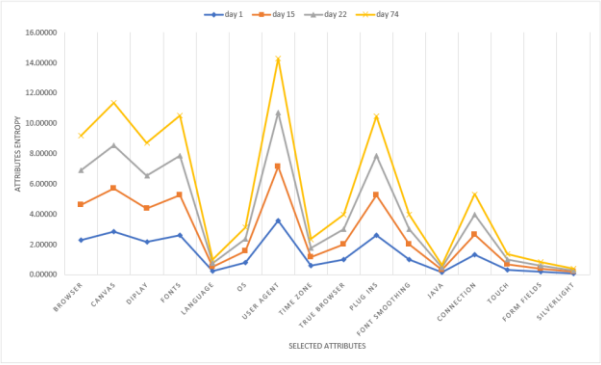

Fig. 2. Attributes changes retrieved by the Chrome browser.

The results shown on Fig. 2 are based on the computer device setup from computer laboratories having the same specifications.

TABLE 3
RESULTS OF ATTRIBUTES CHANGES IN DAYS

ATTRIBUTES	DAY 1	Change	DAY 15	Change	DAY 22	Change	DAY 74	Change
BROWSER	2.29993	0%	2.29993	0%	2.30064	3%	2.30308	14%
CANVAS	2.84447	0%	2.84447	0%	2.84447	0%	2.84447	0%
DIPLAY	2.17889	0%	2.17889	0%	2.17889	0%	2.17889	0%
FONTS	2.62684	0%	2.62684	0%	2.62684	0%	2.62684	0%
LANGUAGE	0.25496	0%	0.25496	0%	0.25496	0%	0.25496	0%
OS	0.78418	0%	0.78418	0%	0.78418	0%	0.78418	0%
USER AGENT	3.56830	0%	3.56830	0%	3.57142	9%	3.57065	7%
TIME ZONE	0.58705	0%	0.58705	0%	0.58705	0%	0.58705	0%
TRUE BROWSER	0.99791	0%	0.99791	0%	0.99791	0%	0.99791	0%
PLUG INS	2.62151	0%	2.62151	0%	2.62151	0%	2.62151	0%
FONT SMOOTHING	0.99791	0%	0.99791	0%	0.99791	0%	0.99791	0%
JAVA	0.15614	0%	0.15614	0%	0.15614	0%	0.15614	0%
CONNECTION	1.32974	0%	1.32974	0%	1.32974	0%	1.32974	0%
TOUCH	0.33660	0%	0.33660	0%	0.33660	0%	0.33660	0%
FORM FIELDS	0.20661	0%	0.20661	0%	0.20661	0%	0.20661	0%
SILVERLIGHT	0.09850	0%	0.09850	0%	0.09850	0%	0.09850	0%
ENTROPY	21.8895	0%	21.8895	0%	21.8934	2%	21.895	3%

This result of attributes changes having 3% entropy difference from day 1 to day 74.

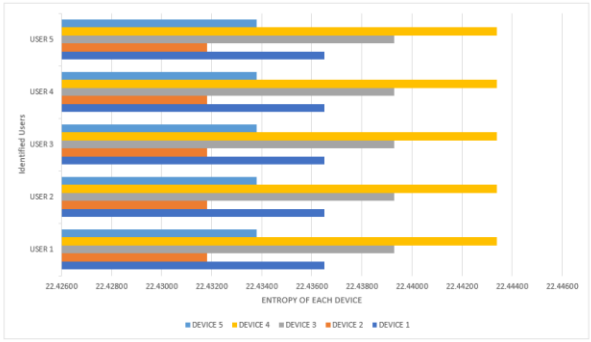

Fig. 3. This result covered within 30 days in evaluating returning devices.

The result shown in Fig. 3 are based on the selected 5 users having different devices configurations. All users are required to login using their account to demonstrate how accurate the identification online. This result provides 100% device identification even if the users are trying to login from other devices.

4 CONCLUSIONS

Device fingerprinting has been an active research topic in web security, particularly web fingerprinting, in recent years. These methods can be used for a wide range of tasks, such as user access control, web tracking and analytics. In this paper, we established a better way of identifying devices using the browser information with the common attributes. Browser fingerprints are really an amazing way to prove that the device you are communicating is legitimate in terms of security communications. The selection of browser fingerprints attributes is important in this research, we identify the common attributes that is not affected by software deprecation and useful to any browser and retrieve only by JavaScript.We hope that our basis, which is freely presented to other researchers and can easily be extended for further studies and it helps to address issues by providing a means to shed light on web fingerprinting practices and methods.

5 REFERENCES

[1]. A. (University of W. Lerner, A. K. (University of W. Simpson, T. (University of W. Kohno, and F. (University of W. Roesner, “Internet Jones and the Raiders of the Lost Trackers: An Archaeological Study of Web Tracking from 1996 to 2016,” *Proc. 25th USENIX Secur. Symp.*, pp. 997–1013, 2016.

[2]. S. Englehardt and A. Narayanan, “Online Tracking: A 1-million-site Measurement and Analysis,” *Proc. 2016 ACM SIGSAC Conf. Comput. Commun. Secur. - CCS’16*, no. 1, pp. 1388–1401, 2016.

[3]. Y. Cao, S. Li, and E. Wijmans, “(Cross-)Browser Fingerprinting via OS and Hardware Level Features,” *Ndss ’17*, no. March, 2017.

[4]. A. Vastel *et al.*, “FP-STALKER : Tracking Browser Fingerprint Evolutions To cite this version : HAL Id : hal-01652021 F P -S TALKER : Tracking Browser Fingerprint Evolutions,” 2018.

[5]. P. De Boer and D. P. Kroese, “A Tutorial on the Cross-Entropy Method,” pp. 1–47.

[6]. P. Laperdrix, B. Baudry, and V. Mishra, “FPRandom : Randomizing core browser objects to break advanced device fingerprinting techniques.” .

[7]. R. Binns, J. Zhao, M. Van Kleek, and N. Shadbolt, “Measuring third-party tracker power across web and mobile,” *ACM Trans. Internet Technol.*, vol. 18, no. 4, 2018.

[8]. A. Kobusi, K. Pawluczuk, and J. Brzezi, “Big Data Fingerprinting Information Analytics for Sustainability Anna,” 2018.

[9]. P. Laperdrix, W. Rudametkin, and B. Baudry, “Beauty and the Beast: Diverting Modern Web Browsers to Build Unique Browser Fingerprints,” *Proc. - 2016 IEEE Symp. Secur. Privacy, SP 2016*, pp. 878–894, 2016.

[10]. K. Mowery and H. Shacham, “Pixel Perfect : Fingerprinting Canvas in HTML5,” *Web 2.0 Secur. Priv. 20*, pp. 1–12, 2012.

[11]. G. Acar, C. Eubank, S. Englehardt, M. Juarez, A. Narayanan, and C. Diaz, “The Web never forgets : Persistent tracking mechanisms in the wild,” *21st ACM Conf. Comput. Commun. Secur.*, no. Section 4, pp. 1–16, 2014.

[12]. O. Starov and N. Nikiforakis, “XHOUND: Quantifying the Fingerprintability of Browser Extensions,” *Proc. - IEEE Symp. Secur. Priv.*, pp. 941–956, 2017.

[13]. X. Liu, Q. Liu, X. Wang, and Z. Jia, “Fingerprinting web browser for tracing anonymous web attackers,” *Proc. - 2016 IEEE 1st Int. Conf. Data Sci. Cyberspace, DSC 2016*, pp. 222–229, 2017.

[14]. A. Vastel, A. Vastel, S. Prof, R. Rouvoy, and P. Walter, “Tracking Versus Security : Investigating the Two Facets of Browser Fingerprinting To cite this version : Tracking Versus Security : Investigating the Two Facets of Browser Fingerprinting .,” 2019.

[15]. Z. Jia, X. Cui, Q. Liu, X. Wang, and C. Liu, “Micro-honeypot: Using browser fingerprinting to track attackers,” *Proc. - 2018 IEEE 3rd Int. Conf. Data Sci. Cyberspace, DSC 2018*, no. 201636000100038, pp. 197–204, 2018.

[16]. A. Gómez-Boix, P. Laperdrix, and B. Baudry, “Hiding in the Crowd: an Analysis of the Effectiveness of Browser Fingerprinting at Large Scale,” *WWW2018 - TheWebConf 2018 27th Int. World Wide Web Conf.*, p. 10, 2018.

[17]. G. Acar *et al.*, “FPDetective,” *Proc. 2013 ACM SIGSAC Conf. Comput. Commun. Secur. - CCS ’13*, pp. 1129–1140, 2013.

[18]. P. Laperdrix, W. Rudametkin, and B. Baudry, “Mitigating Browser Fingerprint Tracking: Multi-level Reconfiguration and Diversification,” *Proc. - 10th Int. Symp. Softw. Eng. Adapt. Self-Managing Syst. SEAMS 2015*, pp. 98–108, 2015.

[19]. F. Rochet, F. Koeune, K. Efthymiadis, and O. Pereira, “SWAT: Seamless web authentication technology,” *Web Conf. 2019 - Proc. World Wide Web Conf. WWW 2019*, vol. 2, pp. 1579–1589, 2019.

[20]. I. Sanchez-Rola, I. Santos, and D. Balzarotti, “Clock around the clock: Time-based device fingerprinting,”

Proc. ACM Conf. Comput. Commun. Secur., pp. 1502–1514, 2018.

[21]. P. Eckersley, “How unique is your web browser?,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 6205 LNCS, pp. 1–18, 2010.

[22]. A. Vastel, W. Rudametkin, and R. Rouvoy, “FP-TESTER : Automated Testing of Browser Fingerprint Resilience,” *Proc. - 3rd IEEE Eur. Symp. Secur. Priv. Work. EURO S PW 2018*, pp. 103–107, 2018.

[23]. N. Kaur, S. Azam, K. Kannoorpatti, K. C. Yeo, and B. Shanmugam, “Browser Fingerprinting as user tracking technology,” *Proc. 2017 11th Int. Conf. Intell. Syst. Control. ISCO 2017*, pp. 103–111, 2017.

[24]. A. N. P. Rogram and T. Thompson, “EVERYONE IS DIFFERENT,” vol. 50, pp. 27–50, 2017.

[25]. A. Abouollo and S. Almuhammadi, “Detecting malicious user accounts using Canvas Fingerprint,” *2017 8th Int. Conf. Inf. Commun. Syst. ICICS 2017*, pp. 358–361, 2017.