Journal of Theoretical and Applied Information Technology
31st July 2022. Vol.100. No 14

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

DE-ANONYMIZATION OF THE USER OF WEB RESOURCE WITH BROWSER FINGERPRINT TECHNOLOGY

EVGENY KARPUKHIN[1], VADIM SHARMAEV[2], ANTON PROPP[3]

[1]Saint Petersburg National Research University of Information Technologies, Mechanics and Optics University ITMO, Saint Petersburg, Russian Federation

[2]Joint Stock Company "Control Systems", Moscow, Russian Federation

[3]National Research University “Moscow Power Engineering Institute”, Moscow, Russian Federation

E-mail: [1]karpukhinmai@gmail.com, [2]sharmaevv@mail.ru, [3]propp.ant@mail.ru

ABSTRACT

The paper highlights the main characteristics of the user, which can be used in the formation of the browser fingerprint, revealing their features. De-anonymization of the user can be used to create individualized advertising campaigns that match the interests of the person, to improve systems for recommending content (for example, articles, videos and music), for secure authentication, collecting statistics about site visitors and analytics. The article also presents other possible scenarios for applying the technology. The methodology presents three possible scenarios: cross-browser solution, maximum amount of data and high accuracy. For each of them, the most appropriate array of user characteristics used to form the fingerprint is chosen, and examples of the JavaScript script are demonstrated. The disadvantage of the technology is the fact that when we change the value of one of the analyzed parameters, the entire output data block also changes. The solution to this problem is to choose the optimal sensitivity threshold. Calculated the optimal sensitivity threshold depending on the number of analyzed parameters, we give examples of its use to determine whether to consider the web service user as a repeat visitor or a new user.

Keywords: *User de-anonymization, Browser fingerprint, Device fingerprint, Information security, JavaScript.*

1. INTRODUCTION

Browser fingerprinting is a technique used by online services and sites to identify visitors by assigning each user a unique identifier (fingerprint), for example: mhxkbwxa6mrxpz5g. This identifier depends on a set of user parameters, which are a unique array of data, such as a combination of information about the screen resolution, installed fonts and model of the device used [1]. The resulting fingerprint will remain constant even if the user switches to incognito mode or turns on a VPN.

The name of this method defines its key feature: the obtained identifier is unique, just like real fingerprints. Because of its uniqueness, the obtained identifier is also called device fingerprint.

The original use of this technology was to optimize the site for the user, regardless of what device the user visited the online resource: from phone, tablet or computer. Without unnecessary actions, the user will be able to see usual news feed on topics of interest, even not yet logged in to the site, will remain the user settings and specified preferences. The technology has found its application in advertising. So, the server, having collected information about the user's behavior model and his characteristics, can fine-tune a personal (targeted) advertising campaign. Such advertising is more accurate than ads based on a simple analysis of a user's IP-address. Certain characteristics of a device can also be used: for instance, a person with a low screen resolution (1024x768) can become a potential buyer of a new monitor in an online store, while a person who visited a store page in the days following a major release of a new smartphone model he is using can become interested in upgrading his device.

An important role is given to device fingerprints when moderating online resources [2]. An intruder who changed his IP-address and his account will remain blocked, because in addition to these characteristics will be analyzed a lot of additional

ones: his device, browser version, operating system, etc. This approach will minimize attacker activity and separate real site visitors from bots logging in from the same device [3].

Often device fingerprints are also used for analytical purposes: they can easily gather statistics on visitors to the site, for example, to know if there is a significant proportion of users with non-standard screen resolution, for which it is worth developing an adaptive version of the site. Browser fingerprints are also used to track the status of the session and for user authentication.

The code that calculates the user ID is described in JavaScript, the language allows linking functionality with HTML elements (also with Flash and Silverlight, discontinued support) and using them as aids.

2. THEORETICAL BASIS

The method of identifying a user on Internet by device fingerprint has replaced tracking cookies [4]. Cookies are small packets of text files stored on a user's computer and contain data that can provide websites with information to improve the user experience. Such files help the developer and the user of an online service by, for example, storing timestamps for a movie watched by remembering user-specified settings. For the developer, cookies are a tool to collect statistics, to optimize and to improve the site.

However, cookies stored on a user's personal computer can be deleted either through browser settings or manually, which is problematic when using them as a unique user identifier. In contrast, the user's browser fingerprint is stored on the server and is independent of the user's actions.

A variety of information can be used to form the browser (device) fingerprint [5]:

- user-agent (line that includes information about the browser and its version, device type, language settings, etc.);
- time zone (difference in minutes between Coordinated Universal Time (UTC) and local time of the user's device);
- screen resolution and color depth (additionally, screen resolution can provide information about whether the device supports screen rotation);
- installed system fonts (in this case a simple enumeration by adding to the page an item with a font from the array of fonts being checked and checking whether the size of the characters has changed: if the character size has changed, it means that the font is installed in the system);
- installed plug-ins and their versions (despite the fact that modern browsers do not give the entire list of installed plug-ins, again a simple search will suffice: in case a plug-in from the array of checked plug-ins is installed on the user, the browser will confirm this request);
- operating system and a lot of other information, such as the ban on geolocation detection [6, 7], font anti-aliasing, connection type [8], etc.

With the development of JavaScript and the emergence of new browser features, this list continues to expand. A new approach for shaping the browser footprint is Canvas [9].

Canvas is HTML5 tag designed to create a bitmap image using scripts, usually in JavaScript. WebGL uses HTML5 canvas to render 2D and 3D graphics in the browser. The essence of the approach is that each computer renders (draws) the image differently due to the peculiarities of computer configuration, characteristics of the operating system and properties of the browser. The resulting image can be used as a unique identification code, turning it into a hash.

VR4Xu19CZgU1dn1ud0zAzMsw6bsu6AigijqRKO4Rl1kwXyJZnEhLohLXJKoUWPUaKLGJLhERNw
wav5okk/zKUaNEdx3ECGICLIzrAlyMHvX/5zquj3VNdXTVd1VPd09730enWH6rudW31PvehXyvB
gw+gM4EMD+APYBMAhAXwC9AAxOMf3VALYCqALA31cA+ATAIgW1XrcxYFRYfR8AgP+NBlAJoB
xAR+un/r2z1a4aQC2AGus//fsOAIsBLLF+cizWiRcj2HVANa8jz7dQpiclCAJFioDKt3UZMEgUxwP4
KoAjAAwMeI5fAiAJkCB6BNx3ojuy1XOV2PbiKNR9eCQ67zgWXXE0AE1D2Q+8FsDbAF4H8BOF
SE9FWMqDE8Vi6ySmoW8e94BTOTrhQ3qV4sMdlIOESOQF18oA8ZXAjwK4JuWpFGQkL8D4GkA
z1riTotFRAEcDJhEcox1dHQNbKkkEA79NjTiVFIWIZDAMM+0o24ALgNw+UEjO3Wr7Fxi9rOzuhE
fLdtNSfYZADcBWJXpANJOEMgFAm1GlAYMqqCmADgXwH65WGwYY1BPNhvAQwCWZjLAOIs6
z2xFlee/X06FU5oNpTiFpCIF4h/QAFscBGDuIFm9u91w3iAM6UtBuLmsqqrF7DmbcNQDa0gkV1

Figure 1: Image obtained using Canvas tag and its corresponding code

In order to change ID obtained with Canvas, it will not be enough to change the time zone or change the screen resolution. We will need to replace the graphics adapter, the graphics adapter driver, and if there are no graphics adapter and its driver, the entire processor.

Journal of Theoretical and Applied Information Technology
31st July 2022. Vol.100. No 14

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

3. METHODOLOGY

As a part of the experiment, JavaScript script fingerprint.js was executed, the operation of which consists in a sequential call of functions that determine the basic characteristics of the browser and the user's device. All obtained values are written to array and then passed through a hash function [10].

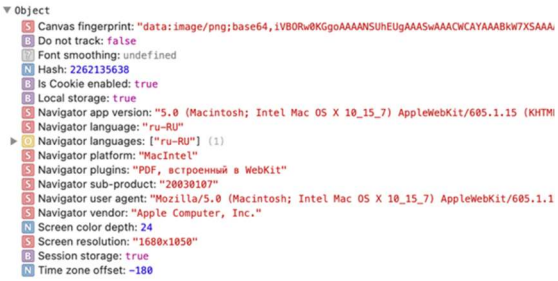

Figure 2: Resulting array of values

The further logic of the script is to check if the database contains a browser fingerprint. If there is a browser fingerprint, it displays a notification that the page has already been visited before. If not, it is added to the database.

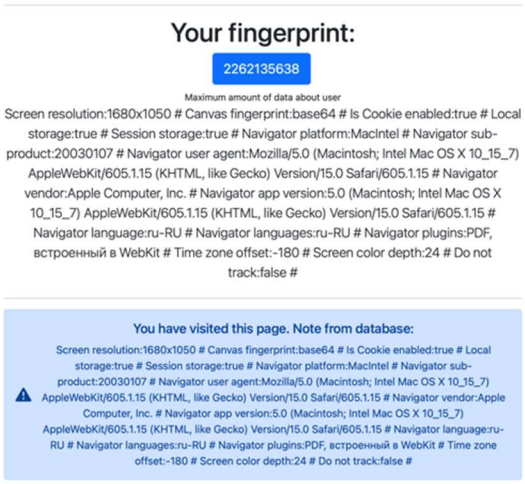

Figure 3: Checking for entries in the database

The experiment was conducted for three possible scenarios:

1. Cross-browser solution.
2. Analyzed parameters: screen resolution, system language, time zone and operating system.

The script is used to de-anonymize the user regardless of the browser it uses. The readout parameters are highly persistent and do not change over time, for example, they will not be affected by a browser update. This approach can be used, for example, to save the user's settings and preferences.

3. Maximum amount of data about the user. Analyzed parameters: screen resolution, Canvas fingerprint, whether cookies are enabled, whether Local Storage is enabled, whether Session Storage is enabled, operating system, browser version, browser, browser developer, system language, plugin list, time zone, color depth, geopositioning prohibition.

The result of the script in this scenario may change regularly regardless of the user's actions, for example, after updating one of the installed plug-ins. In different browsers the result will also vary, but this approach can be used for analytical purposes, for example, in order to determine the target groups of the site.

4. High accuracy. Analyzed parameters: Canvas hash.

The result of the script in this scenario is as stable and accurate as possible, but within a single browser, changing it may require updating the graphics adapter or the entire processor. This approach can be used to control the state of a user's session or to secure an account. For example, when used in online banking services, if the user's browser fingerprint changes, the system may send a confirmation code to another user's device.

For each scenario, the following steps are performed in sequence:

1. Web page is opened in normal mode from Browser №1, and resulting browser fingerprint value is fixed;

2. Browser №1 is launched in private mode (“Incognito” mode), web page is reopened, and obtained browser fingerprint value is fixed.

3. VPN is turned on, web page is opened in the normal mode from Browser №1, and received browser fingerprint value is fixed. VPN is disconnected.

4. Web page is opened in normal mode from Browser №2, and resulting browser fingerprint value is fixed.

5. Browser №2 is launched in private mode ("Incognito" mode), web page is reopened, and the received browser fingerprint value is fixed.

6. VPN is turned on, web page is opened in the normal mode from Browser №2, and received browser fingerprint value is fixed. VPN is disconnected.

7. Current version of Browser №2 is deleted, previous version of Browser №2 is installed, web page is opened normally from Browser №2, and resulting browser fingerprint value is fixed.

The values of all parameters are combined into one string, the resulting string is fed into the hash function MurmurHash3 of the 32-bit version. The result of the hash function is a browser fingerprint.

The algorithm was run on a MacBook Air 13 laptop (MacOS OS) with Safari and Google Chrome browsers installed and on an ASUS R565JA laptop (Windows OS) with Google Chrome and Mozilla Firefox browsers installed. Additionally, and OpenVPN VPN client was installed.

4. RESULTS

During the experiment, the following results were obtained:

1. with a cross-browser solution, the obtained hash remained unchanged when VPN was enabled, incognito mode was enabled, and when the browser was changed. A similar result was obtained on MacOS and on devices with Windows operating system. After repeating the experiment on a different browser version, all devices got the same result;
2. when the maximum amount of data was collected, the hash changed when the browser was changed, but remained unchanged when incognito mode was enabled, and VPN was turned on. A similar result was obtained on MacOS and on devices with the Windows operating system. When repeating the experiment on a different version of the browser, the result changed on all devices;
3. when solving with high accuracy, the hash obtained changed when the browser was changed, but remained unchanged when incognito mode was enabled and VPN was turned on. A similar result was obtained on MacOS and on devices with the Windows operating system. After repeating the experiment on a different browser version, all devices got the same result (Table 1).

5. DISCUSSION

We consider a vector of weights of user parameters $\eta = [\eta_1, \eta_2, \eta_3 \dots \eta_{17}]$ (maximum data scenario, in which 17 different parameters are analyzed), $\eta_1 + \eta_2 + \cdots + \eta_{17} = 1$.

Of the 17 parameters, 7 string parameters are the most unique values of canvas η_1, browser platforms η_6, array of languages η_8, device platform η_9, list of plugins η_{10}, browser version η_{11} and device manufacturer η_{13}). Then we will give these 7 parameters a uniqueness weight of 0.7 and all other parameters a uniqueness weight of 0.3. The resulting vector of weights, taking into account their uniqueness, is the following:

$$I = [0.1, 0.03, 0.03, 0.03, 0.03, 0.1 \dots 0.03]$$
$$I_1 + I_2 + \cdots + I_{17} = 1). \quad (1)$$

The acquisition time and informativity of each of the parameters can be considered the same, then the weight of each parameter, taking into account its acquisition time and informativity, is the following:

$$T = [0.0588, 0.0588, 0.0588, \dots\ 0.0588]$$
$$(T_1 + T_2 + \cdots + T_{17} = 1). \quad (2)$$

We assume that both of these criteria are equivalent, then multiply both vectors by the coefficients of significance of the criteria $\alpha = [0.5, 0.5]$.

Total weight of parameters is the following:

$$\eta = [0.07941, 0.04441, \dots, 0.04441]. \quad (3)$$

We consider an acceptable error, in which only one of the least unique parameters has changed (for example, the parameter "Do not track") η_2,), then the similarity vector $S = [1, 0, 1, 1, \dots 1]$, and the probability of correctly identifying the user is the following:

$$P = S * \eta =$$
$$= 1 * 0.07941 + 0 * 0.04441 + 1 * 0.04441 +$$
$$\cdots + 1 * 0.04441 = 1 - 0.04441 = 0.95559. \quad (4)$$

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Then, taking into account possible rounding, we will take the value of 0.955 as the sensitivity threshold. We consider two more situations:

1) one of the most unique parameters has changed (e.g., canvas parameter η_1):

Similarity vector $S = [0, 1, 1, 1, \dots 1]$. Probability of correct user identification is the following:

$$P = S * \eta = \\ = 0 * 0.07941 + 1 * 0.04441 + 1 * 0.04441 + \\ \cdots + 1 * 0.04441 = 1 - 0.07941 = 0.92059. \quad (5)$$

The value is less than the sensitivity threshold of 0.955. We consider that the web resource was visited by a new user;

2) two of the least unique parameters have changed (for example, "Do not track" parameter η_2 and "Font smoothing" parameter η_3):

Similarity vector $S = [1, 0, 0, 1, \dots 1]$. Probability of correct user identification is the following:

$$P = S * \eta = 1 * 0.07941 + 0 * 0.04441 + \\ + 0 * 0.04441 + \cdots + 1 * 0.04441 = \\ = 1 - 0.04441 - 0.04441 = 0.91118. \quad (6)$$

The value is less than the sensitivity threshold of 0.955. We consider that the web resource was visited by a new user.

Theoretical values are close to the statistical values presented by the research resource Panopticlick, according to which only 1 of 286 777 browsers will give the same fingerprint as the browser of another user. On average, the accuracy of identifying a user with a browser fingerprint is 99.24%. Changing one of the browser settings reduces the accuracy of user identification by only 0.3% [11].

The main variables that determine the reliability of the proposed study are the type of device used (smartphone, tablet, laptop, etc.) and its characteristics, the operating system used (Mac OS, Windows, Linux, etc.) and its characteristics, as well as the browser used and its characteristics. Changing any parameter in the study does not affect the reliability of the result, since the result is determined not by a specific parameter, but by a holistic combination of all analyzed parameters, and most importantly, by its uniqueness. For example, a browser can prohibit determining the user's screen resolution, but the very fact of such a prohibition will also be a distinguishing feature of the user, respectively, the resulting fingerprint will remain unique (with a sufficient number of analyzed parameters). Over time, the browser version may change, and then the generated fingerprint will also change, but the sensitivity threshold will allow detecting the relation of the new fingerprint with the user's previous fingerprint and update it.

The results of the experiment confirmed that this technology has the flexibility to be configured, and with the right choice of parameters analyzed, it also has high accuracy. With increasing parameters the probability of changing the result increases, so it is important to choose the optimal sensitivity threshold, which determines whether to consider a visitor with a changed parameter as the same user or to associate it with a new user.

6. CONCLUSIONS

The technology of forming the browser fingerprint is characterized by its flexibility and, with a competent approach to the selection of analyzed parameters, by its high accuracy. Changing any analyzed parameter changes the final result (changing at least one bit of the input data must lead to a change in the value of the entire output block). To improve the technology we introduce a "sensitivity threshold" for each of the parameters, and the greater the uniqueness of the parameter, the higher its sensitivity (such hashing is called phasiching or fuzzy hashing). In this case the value of the sensitivity threshold depends on the web resource on which it is used. The developer chooses the optimal value, which is a compromise between the situation of excessive false positives and the situation of overreaction of the system.

The accuracy with which the user can be identified in the analysis of 17 user parameters was analyzed. There are off-the-shelf open-source libraries that allow developers to generate and process user browser fingerprints, guaranteeing accuracy in excess of 99.5% at no additional development cost. The use of such libraries can prevent user account theft and fraud. If the unique user ID has changed, it would be reasonable not to block the user, at least to send a confirmation code to an email or phone number or even to terminate the current session.

The question remains how to act if, even with a sufficient number of parameters, the fingerprint of two different users is the same and the web resource makes a false decision that this is the same user. The threshold of sensitivity in this case is not able to solve the problem, since the similarity vector of the characteristics of two users also coincides. In this case, the technology can be improved through auxiliary checks, for example, using the client's repository. When generating a browser fingerprint, the result can be stored on the server with an additional tag (for example, the exact time it was first generated), which is also stored in user cookies with a long shelf life. During subsequent visits to the web resource, in addition to checking for a match between the generated fingerprint and the fingerprint stored on the server, we can also check the data contained in the cookie. If the fingerprint and the timestamp match, we consider that the user has been identified reliably. If the fingerprint matches, but the timestamp is missing or different from the one in the database, we can assume that data about this user is not yet in the server database, despite the presence of a similar browser fingerprint. However, there are other ways to solve the problem, which are determined by other technologies.

On the part of developers, in addition to improving the user interface, the most important task is to ensure its safety. The technology described in the article, combined with other tools, helps to protect user accounts from theft, prevent bank card fraud and protect copyrights. Any actions of an intruder using data of the victim user but possessing a different device fingerprint can be stopped by the system and then other information can be requested: for example, the code word specified during registration on the site. In this case, a security alert is sent to the original device with a recommendation to check the activity in the account and, if necessary, change the password or contact support. The fingerprint can be used to assess the likelihood of fraud or other illegal actions on the part of a particular user.

As any modern technology [12], browser fingerprints can become a dangerous tool for attackers. Avoiding or controlling the collection of browser parameters is almost impossible. The accuracy with which a browser fingerprint identifies a user can become a dangerous tool in the hands of attacker. It is not impossible to further transfer or sell such fingerprint databases, while the users have no way to interfere in this process or influence the use of their data [13].

Identification of users without their knowledge and consent may violate one of the basic principles: right to anonymity. It is almost impossible to exclude the possibility of such parameter collection, the only reliable way to ward it off is to refuse to use Internet [14]. Cookies are currently regulated in a number of countries, and sites must mandatorily ask for consent to process them [15]. Obtaining a fingerprint is an entirely new technology, not yet "touched" by any law in the field of information security. The next logical step could be the creation of regulations for the collection and processing of user characteristics through the formation of browser fingerprints, adopted at the level of legislative acts of countries.

REFRENCES

[1].G. Pugliese, C. Riess, F. Gassmann, and Z. Benenson, "Long-Term Observation on Browser Fingerprinting: Users' Trackability and Perspective", *Proceedings on Privacy Enhancing Technologies*, Vol. 2020, No. 2, 2020, pp. 558–577.

[2].P.A. Ukhov, D.A. Borshchenko, D.D. Kabanov, M.E. Bergen, and A.V. Ryapukhin, "Customization of open-source solutions on the example of the LMS Moodle distance learning platform", *Journal of Physics: Conference Series*, Vol. 1889, No. 2, 2021, p. 022002.

[3].V. Dzhum, and V. Losev, "Review of PI project-based high-level protocol analysis software", *AIP Conference Proceedings*, Vol. 2402, No. 1, 2021, p. 040015.

[4].E. Papadogiannakis, P. Papadopoulos, N. Kourtellis, and E.P. Markatos, "User tracking in the post-cookie era: How websites bypass gdpr consent to track users", *Proceedings of the Web Conference*, 2021, pp. 2130-2141.

[5].A. FaizKhademi, M. Zulkernine, and K. Weldemariam, "FPGuard: Detection and prevention of browser fingerprinting", *IFIP Annual Conference on Data and Applications Security and Privacy*, 2015, pp. 293-308.

[6].M.Y. Klimenko, and A.V. Veitsel, "Evaluation of Neural Network-Based Multipath Mitigation Approach for the GNSS Receivers", *2021 Systems of Signal Synchronization, Generating and Processing in Telecommunications (SYNCHROINFO)*, 2021, pp. 1-5.

[7].A. Podkorytov, "The influence of network structure on quality of satellite corrections for

precise point positioning in GNSS", *IOP Conference Series: Materials Science and Engineering*, Vol. 868, No. 1, 2020, p. 012031.

[8].V.Y. Mikhaylov, and R.B. Mazepa, "USRP devices application for modeling signal-like interference in wireless networks", *2020 Systems of Signal Synchronization, Generating and Processing in Telecommunications (SYNCHROINFO)*, 2020, pp. 1-6.

[9].A. Vastel, W. Rudametkin, R. Rouvoy, and X. Blanc, "FP-Crawlers: studying the resilience of browser fingerprinting to block crawlers", in *MADWeb'20-NDSS Workshop on Measurements, Attacks, and Defenses for the Web*, San Diego, United States, 2020.

[10]. S. Bird, V. Mishra, S. Englehardt, R. Willoughby, D. Zeber, W. Rudametkin, and M. Lopatka, "Actions speak louder than words: Semi-supervised learning for browser fingerprinting detection", *arXiv:2003.04463*, 2020, p. 5.

[11]. C. Hauk, "Browser Fingerprinting: What Is It and What Should You Do About It?", *Pixelprivacy*, 2022, Retrieved from https://pixelprivacy.com/resources/browser-fingerprinting/

[12]. E.V. Vitomsky, and V.Y. Mikhaylov, "Comparative evaluation of the performance indicators of devices for fast sequences delay", In *2019 Systems of Signal Synchronization, Generating and Processing in Telecommunications (SYNCHROINFO)*, 2019, pp. 1-4.

[13]. D.S. Veas Iniesta, and J.G. Estay Sepúlveda, "Development of methods and tools of the commercialization of high-tech projects on the example of Moscow Aviation Institute (National Research University)", *Amazonia Investiga*, Vol. 10, No. 43, 2021, pp 83-95.

[14]. A.A. Povalyaev, "On terms and definitions in radio navigation", *Journal of Communications Technology and Electronics,* Vol. 63, No. 3, 2018, pp. 198-211.

[15]. I.S. Pinkovetskaia, D.F. Arbeláez-Campillo, M.J. Rojas-Bahamón, S.V. Novikov, and D.S. Veas Iniesta, "Social values of entrepreneurship in modern countries", *Amazonia Investiga*, Vol. 9, No. 28, 2020, pp. 6-13.

Table 1: Results of the algorithm on various devices in each of the possible scenarios

Scenario and device	Browser №1, normal mode	Browser №1, private mode	Browser №1, normal mode, VPN	Browser №2, normal mode	Browser №2, private mode	Browser №2, normal mode, VPN	Browser №2, another version, normal mode
"Cross-browser solution", Macbook Air 13	1932166024	1932166024	1932166024	1932166024	1932166024	1932166024	1932166024
"Cross-browser solution", ASUS R565JA	4375241483	4375241483	4375241483	4375241483	4375241483	4375241483	4375241483
"Collecting the maximum amount of data", Macbook Air 13	2262135638	2262135638	2262135638	6132145430	6132145430	6132145430	8462001682
"Collecting the maximum amount of data", ASUS R565JA	5466856785	5466856785	5466856785	6744167049	6744167049	6744167049	4327585695
"High accuracy", Macbook Air 13	1292907387	1292907387	1292907387	6274622209	6274622209	6274622209	6274622209
"High accuracy", ASUS R565JA	7526552456	7526552456	7526552456	5481392048	5481392048	5481392048	5481392048