Advances in Intelligent Systems Research, volume 163

8th International Conference on Management, Education and Information (MEICI 2018)

A Recognition Method for Gradually-Changed Browser Fingerprints

Meilin Zeng[1] and Qiangqiang Xiong[2]

1.Jiangxi Vocational Technical College of Industry Trade Nanchang Jiangxi Province China 330038

2.Nanchang institute of technology Nanchang Jiangxi Province China 330013

Keywords: Browser fingerprint; Static matching; Similarity degree; Gradient fingerprint

Abstract. In recent years, there have been a lot of research on obtaining browser fingerprint and browser fingerprint recognition. Generally, websites use dynamic code embedded in web pages to obtain user display configuration information, system font setting information, browser version and other information, collectively referred to as browser fingerprint information. The fingerprint information of the browser will change with the change of the user's online behavior, so it is impossible to deal with the problem of identifying the same user's browser fingerprint when the fingerprint changes completely by using static identification method. Therefore, this paper proposed a gradually-changed browser fingerprint recognition method. Based on the calculation of character string similarity matching, this method calculated the similarity between browser fingerprint information acquired at different times, and judged whether the two browser fingerprints belong to the same user. Experimental results showed that the proposed method significantly improved the recognition rate of the return visit users, thus verifying the effectiveness of the method.

Introduction

Browsers have become an indispensable tool for people to access the network. People can obtain various network information by sending requests to Web sites. Due to the stateless nature of HTTP protocol adopted by browsers, it is impossible to identify users by the pages they visit. In order to provide users with more accurate and high-quality services, many commercial websites have the need to correctly identify users and their behavior. Therefore, Lou Montulli proposed a user identification method using browser Cookies information in 1994. Browsers can provide users with a privacy operation mode, so the users can protect their own Cookies information, and realize the deletion of Cookies information, therefore, users can avoid exposing their online behavior through the operating mode with the above higher security level, so the accuracy of users identification based on simple Cookies information can not be guaranteed[1-2].

The browser fingerprint produced by the change of using the same user's browser is called the gradually-changed browser fingerprint. Therefore, this paper proposed a gradually-changed browser fingerprint recognition method. Based on the calculation of string similarity matching, this method calculated the similarity between browser fingerprint information acquired at different times, and judged whether the two browser fingerprints belong to the same user. The experimental results showed that the proposed method significantly improved the accuracy of browser fingerprint identification method. Compared with the conventional static fingerprint identification method, the method in this paper was more effective in the identification of return visit users.

Browser fingerprinting and acquisition mode

The main browser fingerprints are shown in Table 1, and the various fingerprint acquisition methods and their meanings are explained.

Table 1 Browser fingerprint information

Order number	Fingerprint name	Fingerprint acquisition method	Fingerprint characteristics
1	Browser type version （useragent）	HTTP, JavaScript	Very easy to change, unstable.
2	http header information （http_accept）	HTTP	The change is small and relatively stable.
3	Plug-in information （plugins）	JavaScript	Upgrading is more frequent and unstable.
4	System font（fonts）	FLASH, JavaScript	The change is small and relatively stable.
5	Display settings （video）	JavaScript	Usually relatively stable.
6	Time zone information （timezone）	JavaScript	The change is small and stable, but the differentiation is not high
7	Cookie setting （cookies）	JavaScript	The change is small and relatively stable.
8	Canvas fingerprint （canvas）	HTML5	The change is small and relatively stable.

The main ways to get browser fingerprints are as follows: One is HTTP. Besides the browser fingerprint of http header information(http_accept), the browser fingerprint of browser type version(useragent) can also be obtained by analyzing the HTTP request information submitted by browser users. The second is JavaScript. If the JavaScript code is inserted into the user's accessing web page, in addition to the browser fingerprint of plugins, the user can also get browser fingerprint features such as timezone, video, cookies and so on. The third is FLASH. The browser fingerprint feature of system fronts can be got by using FLASH. The fourth is HTML5. The browser fingerprint feature of canvas can be obtained through this method.

These browser fingerprint features not only contain the relevant information of the user's browser's application environment, but also are closely related to the user's Internet behavior habits. Therefore, the combination of several fingerprint features can distinguish the individual users who use the browser, thus realizing the identification of browser users.

Change Analysis of Fingerprint Features

This paper analyzed the change rate of browser fingerprints in a certain time through experiments. The number of selected users was 123, and their browser fingerprints were collected separately, and the users were tracked by the cookie information provided in the fingerprint feature Cookie setting (cookies) to obtain the users that can be used for the return visit. Among them, 33 users successfully completed the return visit, and then the browser fingerprints of these users were collected, and fingerprint transformation of the browser was analyzed. By comparing the browser fingerprints twice of the return visit users, the changes are summarized as shown in Table 2.

Table 2 User browser fingerprint change rate

Return visit interval (unit: day)	Return visit user	Changing user	change rate（%）
0～5	32	7	21.9
5～30	8	3	37.5

As can be seen from Table 2, the rate of change varies with the difference of the acquisition intervals of the two browser fingerprints. The change of any fingerprint in the browser fingerprint will result in changes in the entire browser fingerprint, so when the acquisition interval between two browser fingerprints is about 5 days, the change ratio of the user browser fingerprint is 21.9%, and when the acquisition interval between two browser fingerprints is about 5~30 days, the change ratio of the user browser fingerprint is37.5%. Thus it can be seen that, for the browser fingerprint changes of the same user at different times, the static fingerprint matching method can not recognize the changes of browser fingerprint, so it will produce wrong identification results, and will identify the users who have changed the browser fingerprint as new users, resulting in the decrease of the accuracy of user identification.

User Identification Method Based on Gradually-Changed Fingerprint

In order to solve the problem of user's browser fingerprint change, a gradually-changed browser fingerprint recognition method is proposed in this paper. The implementation steps of this method are as follows: Firstly, Hash comparison is done. The objects are the user browser fingerprints collected and the user browser fingerprints in the database. If there is a matching user, it is output as a recognition result. If there is no matching user, the stable fingerprint feature item in the collected browser fingerprints are compared with the stable fingerprint feature item of the browser fingerprints in the database respectively. If there is a matching item, the other gradually-changed fingerprint feature items are compared. If the similarity of the gradually-changed fingerprint feature item is not lower than the set threshold, the browser fingerprint can be considered as a new fingerprint produced after the same user browser fingerprints change, thus realizing the same user's individual identification.

The specific description of the algorithm is as follows:

(1) Hash comparison

When there is a completely matching user browser fingerprint in the database, this matching item can be found by Hash comparison. The subsequent character string similarity calculation is necessary, so the efficiency of user individual identification can be improved.

In this paper, the Hash comparison is implemented based on the MurmurHash operation. Firstly, the collected user browser fingerprints are converted into non-encrypted harsh functions by MurmurHash operation, and then the Hash values of the browser fingerprints stored in the database are compared with the calculated Hash values one by one. If a completely matching Hash value is found, it indicates that the corresponding browser fingerprint and the collected browser fingerprint belong to the same user, and then the individual identification of the user is completed, and the algorithm is executed, otherwise, it goes to step (2) to continue execution.

(2) Stable fingerprint term comparison

The object of the stable fingerprint item comparison is the stable fingerprint feature item that the browser fingerprint contains with certain stability, including the following items: http header information(http_accept), system fonts(fonts), display settings(video), timezone, Cookie setting(cookies), and canvas fingerprint(canvas). The purpose of this step comparison is to narrow the scope of subsequent gradually-changed fingerprint item comparison.

In this paper, the Hash values of the above six types of stable fingerprint items are calculated respectively, and then the Hash values of the browser fingerprints in the database are compared with the calculated Hash values one by one. If the completely matching item cannot be found, the collected user browser fingerprint is saved to the database as a new user. If a matching item can be found, it moves to step 3.

(3) Comparison of gradually-changed fingerprint items

The unstable fingerprint features contained in the browser fingerprints include plug-in information(plugins) and browser type version(useragent). The instability of plugins is caused by

adding, updating and deleting plugins. The instability of browser type version (useragent) is mainly caused by adding, updating and deleting software. Therefore, the fingerprint character strings before and after the transformation have some similarity, and the similarity between browser fingerprints can be measured by calculating the similarity of the character strings before and after the transformation.

In this paper, LevenshteinDistance algorithm is used to calculate the similarity of character strings, which measures character string similarity based on the edit path between the two. The smaller the value, the higher the similarity. The calculation formula is shown in (1).

$$R = \partial_1 \beta_1 + \partial_2 \beta_2 \quad (1)$$

Among them, ∂_1 represents the similarity calculated by plugins, ∂_2 represents the similarity calculated by the useragent, β_1 represents the weight value given to plugins, β_2 represents the weight value given to the useragent, and R represents the ultimate similarity calculated.

The weights given to the fingerprint feature items are proportional to the stability of the fingerprint, so the greater weights are given to the fingerprint items with higher stability. The useragent is more stable than plugins, therefore, it is given greater weight value. The experiment confirms that the weight value given to the useragent is 0.7, and the weight value given to the plugins is 0.3.

The similarity R of gradually-changed fingerprint features between the browser fingerprint in database and collected browser fingerprintis calculated, and the obtained maximum similarity is recorded as $R_{\max}$. When it exceeds the set similarity threshold θ, the browser fingerprint changes gradually and it is used to replace the user browser fingerprint stored in fingerprint database. Otherwise, the browser fingerprint belongs to the new user and is saved to the fingerprint database.

The similarity threshold changes with the access time interval. It is found that only one or two of the fingerprint feature items contained in the user's browser fingerprint change when the return visit interval is short(about 5 days), thus, the browser fingerprint keeps about 98% similarity before and after the change. When the return visit interval is long (about 10 days), many of the fingerprint feature items contained in the user's browser fingerprint change, so that the similarity of the browser fingerprint before and after the change is reduced to about 96%. When the return visit interval is more than 10 days, the similarity of the browser fingerprint before and after change is reduced to nearly 94%. Therefore, the similarity threshold θ set in this paper is shown in Table 3.

Table 3 Similarity threshold setting

Return visit interval(unit: day)	0～5	5～10	>10
Thresholdθ (unit:%)	98	96	94

The flow chart of the gradually-changed browser fingerprint system recognition based on this method is shown in Fig.1.

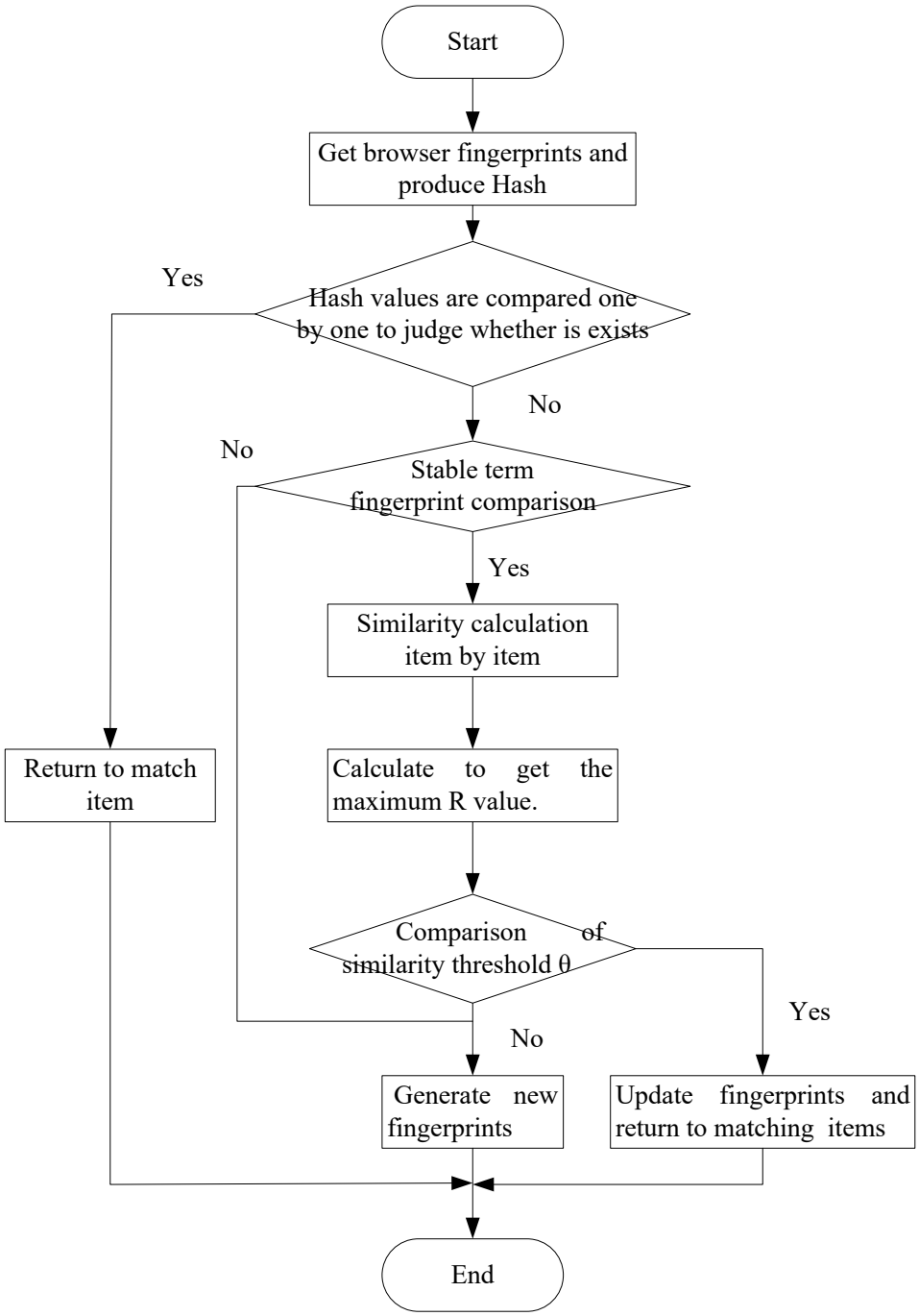

Figure 1. Flow chart of gradually-changed browser fingerprint recognition

Analysis of Experimental Results

In this paper, a special website was built to collect the users' browser fingerprints. The collection time was 15 days, and the number of users was 123. Based on the above data, the performance of the gradually-changed browser fingerprint identification method was verified.

For the individual identification of the return visit users, the static fingerprint identification method and the identification results of this method are compared as shown in Fig.2. The horizontal axis shows the number of days between the last visit and the current visit, and the vertical axis shows the correct proportion of the return visit users identification. The curve in the figure shows the change of the

recognition rate of the return visit users with the access interval, among which, the circular curve uses the static identification method, the triangular curve uses the gradually-changed fingerprint identification method.

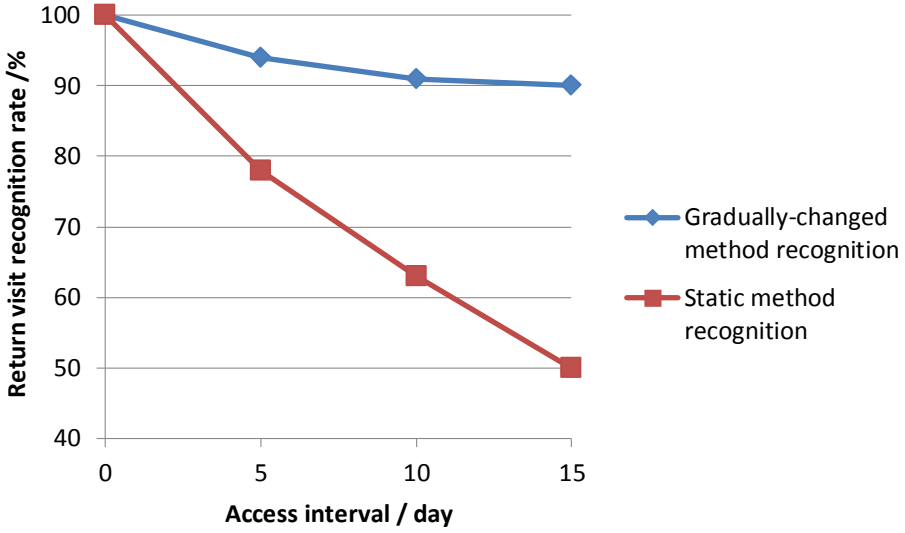

Figure 2. Comparison of the recognition rate of return visit users

The longer the interval between two visits, the greater the probability of the user's browser fingerprint changes. The static identification method can not deal with the change of user's browser fingerprint, so the user identification rate decreases with the increase of time interval. When the user access interval reaches about 10 days, the recognition rate of this method decreases to nearly 60%. This method can still achieve good recognition effect in view of the change of user's browser fingerprints. Compared with the static identification method, the user identification rate of the proposed method is significantly improved.

Conclusions

This paper proposes a browser gradually-changed fingerprint recognition method. By calculating the similarity between the collected browser fingerprints of two visits, the correct recognition of the same browser fingerprint after gradual change can be achieved. A prototype system is built to collect user's browser fingerprints. The validity of the method is verified by taking the data collected as input, which shows that it can significantly improve the user identification rate, not only reduce the identification error, but also increase the number of user fingerprints.

References

[1] Yokoyama S, Uda R.A proposal of preventive measureof pursuit using a browser fingerprint[C]//Proceedingsof International Conference on Ubiquitous Information Management & Communication, 2015: 1-7.

[2]Upathilake R, Li Y, Matrawy A.A classification of Webbrowser fingerprinting techniques[C]//Proceedings ofInternational Conference on New Technologies, Mobility and Security, 2015.

[3] Acar G, Eubank C, Englehardt S, et al.The Web neverforgets: Persistent tracking mechanisms in the wild[C]//Proceedings of ACM Conference on Computer and Communications Security, 2014: 674-689.

[4]Nikiforakis N.Cookieless monster: Exploring the ecosystemof Web- based device fingerprinting[C]//Proceedings ofIEEE Symposium on Security and Privacy, 2013: 541-555.

[5] Takei N, Saito T, Takasu K, et al.Web browser fingerprintingusing only cascading style sheets[C]//Proceedingsof International Conference on Broadband and Wireless Computing, Communication and Applications, 2015: 57-63.
[6]Mowery K, Shacham H.Pixel perfect: Fingerprinting canvasinHTML5[C]//Proceedings of W2SP 2012, May 2012.