Towards Web-based Biometric Systems Using Personal Browsing Interests

Lukasz Olejnik, Claude Castelluccia

Towards Web-based Biometric Systems Using Personal Browsing Interests

Lukasz Olejnik
INRIA
Grenoble, France
Email: lukasz.olejnik@inria.fr

Claude Castelluccia
INRIA
Grenoble, France
Email: claude.castelluccia@inria.fr

***Abstract*—We investigate the potential to use browsing habits and browser history as a new authentication and identification system for the Web with potential applications to anomaly and fraud detection. For the first time, we provide an empirical analysis using data from $4,578$ users. We employ the traditional biometric analysis and show that the False Acceptance Rate can be low ($FAR = 1.1\%$), though this results in a relatively high False Rejection Rate ($FRR = 13.8\%$).**

The scheme may either be utilized by Web service providers (with access to user's browser history) or any Webmaster, using other specialized techniques such as timing-based browser cache sniffing or a browser extension. We construct such a proof-of-concept extension.

I. Introduction

Currently there are on-going efforts to improve the authentication systems[1] and growing needs for new directions. Among the well-known and distinct authentication classes are the ones utilizing *something the user knows* (i.e. password, pin), *something the user has* (i.e. RFID tag, secure token) and *something the user is* or how he behaves - the biometric approach. Biometry is a well-established paradigm used actively in both identification and authentication mechanisms. Most popular biometric solutions are based on the fingerprinting of physical traits such as eye irides and finger's friction ridges. Biometry is often used as a component of a multi-factor authentication system which combines two or more of the authentication classes. *Behavioral* biometrics is an integral part of biometry. It focuses on fingerprinting personal traits such as style of walking, voice, gait etc. Other approaches may consider keyboard typing and mouse moving analyses. The behavioral approach is known to be prone to both *false positives* and *false negatives*, also known as Type I and II errors, respectively, during the fingerprint recognition phase.

As opposed to traditional, physiological biometric fingerprints, behavioral biometric fingerprints (and traits) are subject to change over time. Affected systems must thus take this property into consideration, and frequently update them. However, this does not necessarily pose a problem, especially since behavioral biometrics can be extracted periodically, and thus updated; this being among the differences between physiological and behavioral biometrics. Additionally, behavioral approach can be of use in more specialized environments. Examples of such environments include Web-based systems, where the use of physiological biometry is complex. Applications on the Web typically utilize standard authentication schemes based solely on *something the user knows* (password). This approach certainly has its merits, but improvements addressing various threats and challenges, such as inability to memorize an efficient password or the difficulty to detect anomalies, such as account hijack, are worth investigating.

This paper proposes an authentication scheme that relies on users' interests that are related to the browsing habits. A fundamental tenet of our work is the presumption that resources available in the browser's cache serve as a source of information about the user. Browsers keep the browsing information for usability purposes; most browsers store information about URLs entered into the browser's address bar, a list of all visited Websites and downloaded files. Such data is collectively known as the *browser history*. As such, the fact that a user visited the website of a certain bank, political organization, or a product page at an e-merchant's site will reveal some information about that user. The described system is meant to extend the currently used solutions and serve as an identification aid and potential additional factor in authentication. Therefore it can be a part of a multi-factor authentication system. The addition of the interest-based biometric approach potentially opens interesting possibilities, such as detection of fraud and anomaly analyses, for example in the case of a stolen and misused password: a site may discover whether the browsing history of a given user is consistent with previously-seen ones. Our work touches on the potential applications of cache-sniffing methods, also called history or cache hijacking, for determining users' browsing history This in turn allows either authorization or checking the validity of a query from the user, such as a request for authorization using other, perhaps password-based scheme, or a request in a banking system that could potentially result in a malicious transfer in case of a password or a session cookie loss. However, currently it is not clear what would be the practicality of a solution based on these techniques, mainly due to the performance and operation issues. To address this, we introduce another approach based on browser extension. This mode of operation is both efficient and reliable while maintaining simplicity.

To our knowledge this is the first work exploring this particular approach and specifically these areas of users' interests.

The importance of this work is strengthened by a very recent commercial story of Drawbridge [2], a company aiming to pair the visits coming from the desktop and

mobile users.

In Section II, we evaluate and discuss the biometric potential of personal browsing habits. In Section III, we describe a proof-of-concept solution discussing the requirements, performance and security. In Section III-C, we mention certain special cases and potential applications.

A. Related work

To construct a biometric system, a fingerprinting approach is required. Here we refer to the Web-related approaches and studies of fingerprinting. Fingerprinting on the Web is possible using the JavaScript-accessible browser configurations, as shown in a large experiment by Eckersley [3], where the fingerprinting is based on plugins, fonts and others. Fingerprinting potential based on the detection of browsers' configuration using JavaScript is also analyzed by Mowery et al. [4]. Other important example is [5], where the authors study a large data sample from users of Hotmail and Bing focusing on the potential of tracking relating only to the host information, including browser cookies and User-Agent string.

Behavioral biometry, where fingerprints are based on the behavioral aspects and traits such as typing dynamics or voice analysis is described in [6], [7], [8]. Behavioral biometry on the Web has already been studied, for example using mouse movements [9], [10], [11] and keystroke dynamics [12], [13]. The most advanced work and results from the mouse dynamics domain is perhaps the work by Zheng et al. [14], where the authors obtain good metrics in terms of low false accept and reject ratios. Other behavioral systems may utilize signatures analyses [15]. Individuality of fingerprints is obtained mainly due to empirical studies on available samples, as described in [16].

Identifying the users as human beings is often solved by the use of CAPTCHAs [17]. There are a number of other possible authentication methods and one interesting example may be Facebook, where a user is presented with people's pictures and he is required to choose his friends, as an additional security layer [18]. A good example of anomaly detection is Gmail, which keeps track of originating source IP of the visitors: if the system detects an "unusual" event, a connection from an "unexpected" country, the user might be informed about this fact.

The potential of using Web preferences as another layer was also publicly hinted at, for the first time, in [4]. In this paper we study these concepts in detail.

Our work may be considered an example of service utilization [19], where the user is authenticated to a particular service based on the services usage patterns.

II. Biometric Analysis of Web Preferences

To analyze a biometric system certain common aspects are typically studied [20]. First of all, the fingerprint, meant as a *unique identifier*, should be unique among the users. Though in behavioral biometric systems one normally does not expect the performance of physiological biometric systems, such as the one basing on irides or finger's friction ridges, and this requirement is relaxed. Behavioral-based systems normally utilize a confidence estimation functions, rather than a template based on physical traits.

In a biometric system, *False Reject Ratio* (FRR, Type I errors) is the rate of the system's inability to recognize a legitimate user, a parameter related to usability. *False Accept Ratio* (FAR, Type II errors) is the system's rate of incorrect recognition of a different user (impostor) as a legitimate user; because this is a strictly security parameter, in real systems this error is required to be low. *Equal Error Rate* is a point with same error rate of FAR and FRR and is a rate of the systems' performance. The lower the EER, the better. The typical requirements for a practical biometric system focus also on the functional aspects such as

- *Uniqueness* is a requirement that the trait of choice should be individually-attributable to a sufficient manner; in biometric systems, this factor is typically discovered empirically.
- The system is required to be *accurate*, thus FAR parameter should be as small as possible, but typically a trade-off needs to be made. If the system correctly detects impostors it must not cause any problems with identifying a true user therefore the FRR parameter is also very important.
- *Speed* is an important parameter, the user should not be required to wait for too long on the systems' decision.
- The system is required to be hard to overcome in the case of a users' profile leakage; thus it should be *resistant to circumvention*

The rest of this section presents some of our experimental results on the uniqueness of web histories, and their accuracy.

A. Experimental Setup

The dataset of the actual users' partial Web histories was obtained from the authors of the *What The Internet Knows About You* experiment [21]. This project gathered between 1.09.2009 and 15.05.2011 more than $440,000$ profiles of unique users.

In this paper we refer to the $382,269$ users who executed the "popular links" test, which was testing for over 6000 first-level links. The "popular links" list has been created out of 500 most popular links from Alexa [22], $4,000$ from Quantcast [23] and some random pages from the other tests (e.g. "government and military sites") in the system. The choice has been made this way due to demonstration and education functions. Therefore, with the use of this project (and the browser history issue as well) it was not possible to obtain the whole history. However, this was not necessary, as it is sufficient to show that the actual subsets are unique: a considerable unique number of profiles within a large dataset most likely indicates the overall uniqueness of the whole history superset of these Web users (if the subsets of some

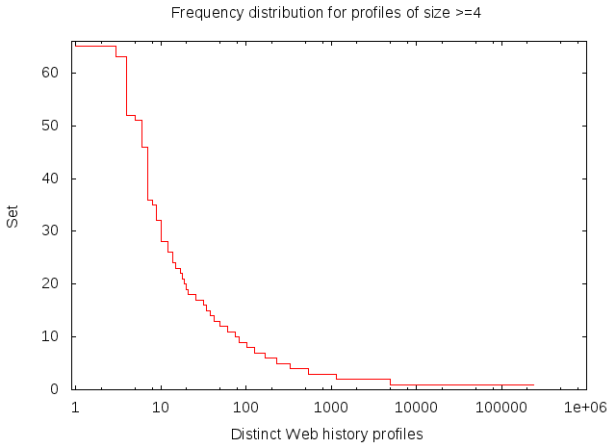

Figure 1: Frequency distributions for profiles of size 4 and larger.

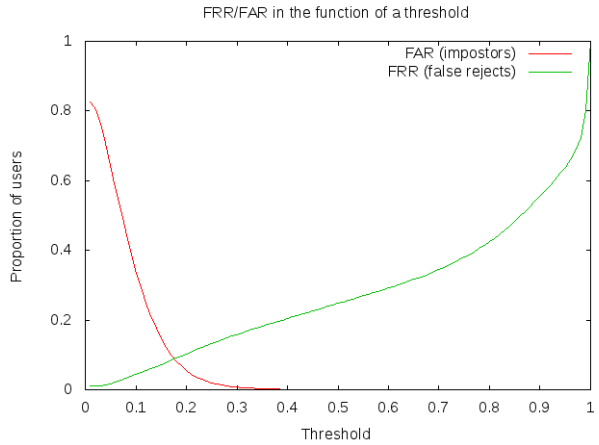

Figure 2: Type I and Type II errors in function of the threshold. Equal Error Rate (EER) is observed at the intersecting point. FRR plot made for the same data of $4,578$ users. FAR is computed for $4,578$ users against $242,805$ different users ("impostors") taken from the whole dataset.

supersets are unique, those supersets must consequently be unique).

The systems' crucial aspect had an educational content: provide the user with information on which (and how many) pages have been successfully detected, and inform him of the problem and the risks, as well as educate him to defend against this attack. The site did not use of any side-information such as cookies, flash cookies or other persistent storage techniques. As was noted in before, the system detects whether a particular site from a pre-defined sites list is present in the browsers history. So it is not unlikely that the number of detected sites will vary among different users.

B. Uniqueness evaluation

We created a frequency distribution of the uniqueness set for the users in our dataset and it is shown on Figure 1. The X axis represents the number of distinct profiles, as counted from the dataset, that correspond to a specific anonymity set (Y axis), ordered from the largest to the smallest set. For example, the point $(X = 10000; Y = 1)$ indicates that the $10,000$th user is unique - it does not share a fingerprint with any other user.

These results show that if the number of discovered links of a user is at least 4, then profiles are almost certainly unique, with a uniqueness rate 97% in the studied dataset. As was demonstrated in the work of [24], it is only required to test the user's history against a small pre-defined list. This list can be as compact as 500 URLs. Therefore in the biometric analysis we refer only to the profiles of this - or larger - size.

C. Accuracy evaluation

1) Distinguishing Fingerprints: In order to differentiate between the fingerprints, which in our case are sets of visited links, we employ the Jaccard Index. For two sets A, B the Jaccard Index is computed as $\frac{|A \cap B|}{|A \cup B|}$. Two sets are equal if Jaccard Index is 1, and they are highly correlated if it is larger than 0.7. The problem is to choose a good threshold value. Threshold $t = 1$ will require an identical fingerprint which will result in accepting of an ideal fingerprint only: impersonation will be limited, however, the rate of rejection might be too high since the fingerprints may change. On the other hand, if the threshold is low, the impersonation rate can be too large. A trade-off is obviously required.

2) False Accept/Reject Analysis: In order to conduct a False Reject Ratio analysis it is required to have several fingerprints of the same user. For this analysis we have used the data from $4,578$ revisiting users; these users have been fingerprinted several times so a comparison is possible. Additionally, the number of all users with profile size larger than 4 is $242,805$ and consequently we can also use the fingerprints from these users to strengthen our False Accept Ratio analysis by testing the $4,578$ revisiting users against this larger sample.

Figure 2 shows the FAR and FRR curves as a function of the threshold. FRR is computed for every fingerprint of same users from the sample of $4,578$ users.FAR however, is obtained by comparing the fingerprint of a given, selected, user between all the $242,805$ histories of users in our database. Therefore and obviously, the number of pairs for FAR is larger but it is consequently studied against a large sample of unique users. The equal error rate lies close to the threshold value of 0.166 ($FAR = 9\%$, $FRR = 8\%$). However, in reality FAR is very important: the system should not allow an impostor to identify himself as a true user. For this reason a more conservative value of the threshold could be 0.6. The consequence would be almost zero impostor ($FAR = 0.002\%$) rate, but about 29% rate of false rejects. Selecting threshold of value 0.37 results in $FAR = 0.1\%$, thus one in a thousand users will be falsely accepted and 19% falsely rejected. Threshold 0.25 results in $FAR = 1.1\%$ and $FRR = 13.8\%$ which can be a compromise.

Additionally the system could employ a per-person

threshold value to reflect the individual habits potentially influencing on the fingerprints' changing.

3) Receiver Operating Diagram: To analyze the operating performance of biometric systems one typically need to compare the error rates independently of the threshold. The system's performance can be observed on Figure 3, the Receiver Operating Characteristic (ROC) plots the False Reject Rate (FRR, Y axis) as a function of False Accept Rate (FAR, axis X); the two curves are made from the the same data as previously. The curve lies very close to the coordinate axis which suggests the system performs well.

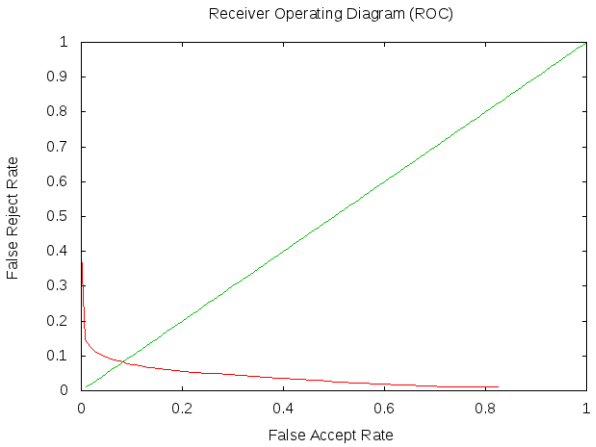

Figure 3: Receiver Operating Characteristic

4) Summary: The systems' robustness has been analyzed using the traditionally-employed biometric analysis. It is observed that not only users' Web preferences are of behavioral biometric potential, but the system can recognize the same users and distinguish them with reasonable efficiency. As a future direction, we believe FAR/FRR analysis could be done using a larger data sample, perhaps from controlled users group, to establish even tighter bounds on the security parameter.

In the end we note that the data-collecting effort has not been done with the biometric analysis in mind and this application is an unexpected outcome of the large uniqueness (about 97% for profiles of 4 and larger) and non-negligible stability within revisits [24]. The idea is strictly based on the obtained data.

III. Interest-based Authentication

As shown previously, users' browsing habits are largely unique and constitute a good biometric candidate. In this section, we describe how users' web histories can be leveraged to authenticate them on the web.

Such system would rely on two steps: the registration and authentication phases:

1) *Registration Phase:* the user registers his signature to the server it wants to authenticate. A signature is a list of sites that identifies him uniquely.
2) *Authentication Phase:* when the user wants to access the service provided by a server, he needs to prove that he knows the signature, i.e. that he has been visiting the sites that composed his signature. Two solutions are then possible. `(1)` The server hijacks the user's history without his consent and then verifies the signature. `(2)` A protocol is run between the user and the server to verify the signature. These two approaches are described in the remaining of this section.

A. Biometric Authentication Using Preference Hijacking

1) Web Preferences: Fingerprint Extraction Techniques: The ability to determine user preferences from the information stored in the browser's history hinges on the capacity to query the browser cache for the availability of resources. Several known techniques exist, which allow for obtaining reliable information about the existence of various kinds of information in the cache, as described in [25] and [26], [27]. The general concept of extraction requires the user to visit a specified site. The site then is able to check the contents of the users' browser cache; namely, whether the user has visited a number of specified sites from a previously-chosen set.

There is a number of different approaches to the extraction of Web preferences data from a browser. Here we describe some possible modes which can be utilized.

1) *History hijack* Originally, it was possible to detect an already visited link on a currently-visited site. This was usually done by different link color being displayed. The problem we are addressing here arises from several key browser technologies. The most important one is JavaScript, which allows this by checking the CSS *:visited* property. In modern browsers this approach is impossible due to a recently introduced fix [28].
2) *Browser cache* Browsers use cache due to performance reasons. Pre-cached elements allow a faster download and rendering of a visited site. This, however, is prone to a *timing analysis*: object in cache is retrieved much faster. The problem with this approach was that testing whether the object is in the cache pollutes it and all subsequent tests will report this particular item in cache - increasing the rate of false positives. Non-destructive timing cache analysis addresses these problems. This approach is demonstrated to work [27].
3) *DNS caching* Operating systems usually maintain a DNS caching mechanism which allow a faster resolution of a network name. This potentially allows the leakage of visited sites, because unvisited sites will require a name resolution, which is often time consuming and thus allows the timing analysis.

2) Protocol Overview: The implementation is straightforward. The authenticating server is of course required to have a previous knowledge about the specific user's interest in order to compare the already-known fingerprint to the currently-seen one. The server uses one of the existing browsing history hijacking attacks to check whether the sites from the user's signature (partial browser history)

are currently in his cache. The problem, however is that any Web page can access this information. In order to address this the server can choose a *per-user fingerprint*: a specific list of test sites for a particular user, this list is required to remain confidential.

We have implemented a timing-analysis based scheme as a simple JavaScript. The system has been randomly selecting the links to test and after that the user was presented with those to be visited. Subsequently, the system authenticated the user only if these right links were detected as visited. Tests have been conducted using a small number of links and only for the Opera browser. They prove this approach is feasible but highlight a possible problem: some of the sites often change the deployed resources and due to these changes the working system is subject to constant updates. The updates can be automated, though.

3) Security Analysis: The problem with this approach is the possible user's impersonation. Every Web page can potentially scan the users' browser's history and hijack the fingerprint (if the sites used for sampling purposes is known). In order to protect against the threat of history sniffing attacks the user may use additional plugins. NoScript extension [29] can block the execution of scripts and maintain a white-list of pages which can execute them. Moreover, the user may be interested in blocking or limiting the use of cross-domain scripting interactions which may be realized by the use of CsFire plugin [30]; this approach would render useless the currently-known timing analysis threats. But on the same time, complicate the use of a timing analysis technique in a legitimate biometric deployment.

If the fingerprint is a per-user list of sites and it remains a secret, it would not be clear for the potential attacker for which sites to test.

This approach could potentially be still valid for two uses:

- **Anomaly detection** – verification of fraud and unusual actions by detecting the real user
- **CAPTCHA** – detecting whether the visitor is a human being

A more secure approach can be obtained by extending the functionality of a Web Browser.

B. Biometric Authentication Using Browser Extension

We have investigated the efficacy of a solution utilizing a browser extension. We have constructed such an extension for the Firefox browser and verified the feasibility of this system in practice. The proof-of-concept's purpose is to send the users' Web history to the server in order to identify or authenticate the user. We assume here that the fingerprint is composed from the sampling of a users' browser's history. For simplicity we sample the history considering only the 500 most popular Web sites from Alexa ranking [22]. The systems' functionality need to cover: the registration and authentication phases. Registration step is done when the user is visiting the specified site for the first time: the server records a currently seen fingerprint. This fingerprint is used later during authentication.

The user generates a seed that is secret and stored in the plugin. This operation is done only once, for example before the first use. For the service S the user is prompted for an access password p_S to this service and the plugin computes, for each service, a salt $salt_s = hash(seed, p_S)$ from seed and password. After this, the plugin computes a HMAC $h_{s,i} = h(salt_s, site_i)$ for every $site_i$ in the users' web history. The presence of a $site_i$ (for certain $1 < i < 500$ in our case) in the browser's history means the user has visited this site. The $fingerprint$ set is constructed from such sites of choice; in our case $site_i$ is used if it was both visited and belongs to the 500 most popular Alexa sites. In the end the set $fingerprint_{s,h} = \{h_{s,i}, ..., h_{s,n}\}$ is composed from the hashes of the visited sites. The $fingerprint_{s,h}$ is then provided to the service during the registration phase.

Upon authentication to a service over HTTPS, the user is prompted with the service password by the plugin. The plugin then computes the keyed HMAC using the sites present in the user's web history, the seed and the entered password. The generated fingerprint is then sent to the server together with the password p_S. The server then can compare the provided fingerprint to the previously registered one Additionally, the server can interpret the possible slight changes in the previously-known fingerprint versus the currently seen one. The $fingerprint$ can be tailored per service. Thus, different Web sites or services can sample the browser history in search for different sites.

The described scheme is a behavioral 2-factor authentication system that is based on "something the user knows" (the service's password) and the user's browsing history; the user needs to know the system's password p_S. Additionally user is required to have a device in form of a browser with a dedicated plugin that contains the initial $seed$ and to previously visit Websites according to the user's preferences. The extension for the Firefox browser is currently available at http://www.inrialpes.fr/planete/people/lukasz/wprefbiom3.xpi.

The proof-of-concept based on browser extension is efficient; it is fast and easy to retrieve the users' history. Furthermore, the users' privacy is preserved: never during the execution of this protocol the server discovers the actual history contents due to the use of a hash function. The disadvantage is that the plugin consists of a 3rd party software that has to be installed in order to leverage this system.

1) Security Analysis: The fundamental advantage of this system is that the server does not learn the users' history, thus his privacy is preserved. However, what about the security risks? There are at least several vectors of attack to hijack the history. One prominent is the Web history sniffing attack [25] and more recently timing-analysis approaches had been unveiled [27]. Using both of these techniques it is possible to hijack the users' history with high accuracy. But in the browser plugin case these risks are limited due to the additional use of a per-plugin

seed and the actual services' password to create the list of hashed URLs in the browser's cache. Additionally, the systems' performance is unaffected by the possible attacks utilizing of Web-techniques to hijack the user's history.

Moreover, the fingerprint can be tailored in a way that a general attack against a particular user can be made infeasible. The service can achieve this by, for example, issuing a per-user list of sites to test for the fingerprinting purposes. Even if a malicious website has stolen the user's history, the security of the systems' operation is maintained.

C. Special Cases and Simplified Schemes Using Browsing Interests

In addition to the previously described schemes, other potential applications might be developed.

1) PIN-like Scheme: The previously described approaches can be easily simplified by requiring a user to visit a number of per-user defined sites just before the attempt to log into the server. This way it is easier to both ensure the presence of sites in the browsers' cache and at the same time keep the system working. Of additional note is that this would defend against device and banking PIN thefts (due to the use of keyloggers, for example), as the previous actions are just based on ordinary Web browsing. There may be a requirement to clear the user's history before entering these chosen sites: this way, the system will correctly detect only the previously-agreed upon sites. Testing can be implemented via either Web techniques or a browser plugin, as previously. The sites used "as a PIN" can be configurable but should remain a secret between the user and the server.

2) The Special Case of Web Service Providers: Since *Web service providers*, such as Facebook and Google, have access to information regarding what their users browse via either 3rd-party scripts (e.g. *Facebook Like*, *Google +1* buttons) or search engine queries (in the case of Google), they could use this data in a simple system. When a user wants to login, in addition to a password requirement, he is presented with a challenge: a number of sites (or categories such as the attribution of a news site Website with a certain category of sites like "News Sites") which he did or did not visit. The user then is required to choose the right answers. The system verifies them and as a result, grants access or not. Among the advantages are: service providers typically have large databases of sites, they have the capabilities to detect what their users browse and no 3rd party application is required. Such mode of operation may resemble the well-known Facebook's social captcha system [18].

3) Anomaly detection: Anomaly detection can leverage the pure Web-based approach. Although every site could still hijack the fingerprint using the same techniques, this approach would offer something which is not currently possible: verify a person based on his personal traits. The mode of operation would employ the known Web based techniques to verify the browser's history content. But rather than deciding on granting or not granting access to a resource, the system would just log any suspicious activities, for example if the fingerprint is vastly different than the usual; or the nature of detected sites is inconsistent with the known ones. The system could trigger an alert, with the potential possibility of blocking such users. As was noted previously, any Web site can access this information and possibly impersonate the user, but since here it is being used solely as an additional information for the hypothesized systems' auditing architecture it does not matter.

The False Accept Ratio in our tested scheme was quite significant and this also hints the use of these-kind of systems to anomaly detection.

IV. Conclusion

In this paper we have described a novel biometric method for the Web. First of all, we have proposed to treat the Web browser habits as behavioral biometric traits. Second, we analyzed this basing on a sample of real-world data from the Web users. The results suggest that the construction of a biometric system taking advantage of these user characteristics is possible. Such solution could be of use in the identification, authentication and anomaly detection domains. The systems' False Accept Rate (FAR) can be low and within the European Standards [31]. However, since this raises False Rejection Rate (FRR) therefore either a trade-off need to be made or a per-user threshold system should be used; which in the case of behavior biometrics is obtainable.

The collection of browsing habits has usually been possible using a history hijack sniffing attack. Even though this issue has been resolved, other approaches, such as timing-analysis techniques or a dedicated browser plugin are still compelling solutions and in fact we have verified the practicality of both of them.

Entities providing Web services are in an especially good position to use systems of these kinds, be that Web, browser plugin or internal data on the sites visited by their users or performed search engine queries. This comes from a fact that Web service providers usually can have an access to means allowing the extraction of user profiles, the visited sites of the users, in an efficient manner.

References

[1] A. A. E. Ahmed and I. Traore, "Darpa seeks authentication beyond passwords."

[2] K. D. Vere, "Drawbridge exits closed beta, unveils cross-screen mobile marketing solutions." [Online]. Available: http://www.insidemobileapps.com/2012/11/15/drawbridge-exits-closed-beta-unveils-cross-screen-mobile-marketing-solutions/

[3] P. Eckersley, "How unique is your web browser?" in *Privacy Enhancing Technologies*, 2010, pp. 1–18.

[4] K. Mowery, D. Bogenreif, S. Yilek, and H. Shacham, "Fingerprinting information in javascript implementations," in *Proceedings of W2SP 2011*, ser. IEEE Computer Society, 2011.

[5] T.-F. Yen, Y. Xie, F. Yu, R. P. Yu, and M. Abadi, "Host fingerprinting and tracking on the web:privacy and security implications," in *19th Annual Network and Distributed System Security Symposium (NDSS) 2012, Internet Society*, 2012.

[6] R. Moskovitch, C. Feher, A. Messerman, N. Kirschnick, T. Mustafic, A. Camtepe, B. Löhlein, U. Heister, S. Möller, L. Rokach, and Y. Elovici, "Identity theft, computers and behavioral biometrics," in *Proceedings of the 2009 IEEE international conference on Intelligence and security informatics*, ser. ISI'09. Piscataway, NJ, USA: IEEE Press, 2009, pp. 155–160. [Online]. Available: http://dl.acm.org/citation.cfm?id=1706428.1706455

[7] B. Miller, "Vital signs of identity," *IEEE Spectr.*, vol. 31, pp. 22–30, February 1994. [Online]. Available: http://dl.acm.org/citation.cfm?id=187762.187765

[8] R. V. Yampolskiy and V. Govindaraju, "Behavioural biometrics: a survey and classification," *Int. J. Biometrics*, vol. 1, pp. 81–113, June 2008. [Online]. Available: http://dl.acm.org/citation.cfm?id=1389509.1389515

[9] C. Shen, Z. Cai, X. Guan, H. Sha, and J. Du, "Feature analysis of mouse dynamics in identity authentication and monitoring," in *Proceedings of the 2009 IEEE international conference on Communications*, ser. ICC'09. Piscataway, NJ, USA: IEEE Press, 2009, pp. 673–677. [Online]. Available: http://dl.acm.org/citation.cfm?id=1817271.1817397

[10] R. Moskovitch, C. Feher, A. Messerman, N. Kirschnick, T. Mustafic, A. Camtepe, B. Löhlein, U. Heister, S. Möller, L. Rokach, and Y. Elovici, "Identity theft, computers and behavioral biometrics," in *Proceedings of the 2009 IEEE international conference on Intelligence and security informatics*, ser. ISI'09. Piscataway, NJ, USA: IEEE Press, 2009, pp. 155–160. [Online]. Available: http://dl.acm.org/citation.cfm?id=1706428.1706455

[11] M. Pusara and C. E. Brodley, "User re-authentication via mouse movements," in *Proceedings of the 2004 ACM workshop on Visualization and data mining for computer security*, ser. VizSEC/DMSEC '04. New York, NY, USA: ACM, 2004, pp. 1–8. [Online]. Available: http://doi.acm.org/10.1145/1029208.1029210

[12] F. Monrose and A. Rubin, "Authentication via keystroke dynamics," in *Proceedings of the 4th ACM conference on Computer and communications security*, ser. CCS '97. New York, NY, USA: ACM, 1997, pp. 48–56. [Online]. Available: http://doi.acm.org/10.1145/266420.266434

[13] S. Giroux, R. Wachowiak-Smolikova, and M. P. Wachowiak, "Keystroke-based authentication by key press intervals as a complementary behavioral biometric," in *Proceedings of the 2009 IEEE international conference on Systems, Man and Cybernetics*, ser. SMC'09. Piscataway, NJ, USA: IEEE Press, 2009, pp. 80–85. [Online]. Available: http://dl.acm.org/citation.cfm?id=1732323.1732337

[14] N. Zheng, A. Paloski, and H. Wang, "An efficient user verification system via mouse movements," in *Proceedings of the 18th ACM conference on Computer and communications security*, ser. CCS '11. New York, NY, USA: ACM, 2011, pp. 139–150. [Online]. Available: http://doi.acm.org/10.1145/2046707.2046725

[15] V. S. Nalwa, "Automatic on-line signature verification," *Proceedings of the IEEE*, vol. 85, no. 2, pp. 215–239, 1997. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=554220

[16] S. Pankanti, S. Prabhakar, and A. K. Jain, "On the individuality of fingerprints," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 8, pp. 1010–1025, Aug. 2002. [Online]. Available: http://dx.doi.org/10.1109/TPAMI.2002.1023799

[17] L. V. Ahn, M. Blum, N. J. Hopper, and J. Langford, "Captcha: using hard ai problems for security," in *Proceedings of the 22nd international conference on Theory and applications of cryptographic techniques*, ser. EUROCRYPT'03. Berlin, Heidelberg: Springer-Verlag, 2003, pp. 294–311. [Online]. Available: http://dl.acm.org/citation.cfm?id=1766171.1766196

[18] A. Rice, "A Continued Commitment to Security ," http://blog.facebook.com/blog.php?post=486790652130, 2011, [Online; accessed 6-Mar-2012].

[19] N. L. Clarke and S. M. Furnell, "Authentication of users on mobile telephones - A survey of attitudes and practices," *Computers & Security*, vol. 24, no. 7, pp. 519–527, Oct. 2005.

[20] S. Prabhakar, S. Pankanti, and A. K. Jain, "Biometric recognition: Security and privacy concerns," *Security & Privacy, IEEE*, vol. 1, no. 2, pp. 33–42, 2003.

[21] A. Janc and L. Olejnik, "What the internet knows about you," http://www.wtikay.com/.

[22] Alexa, "Alexa 500," http://alexa.com.

[23] Quantcast, "Quantcast," http://www.quantcast.com/.

[24] L. Olejnik, C. Castelluccia, and A. Janc, "Why johnny can't browse in peace: On the uniqueness of web browsing history patterns," in *HotPETS*, 2012.

[25] A. Janc and L. Olejnik, "Web browser history detection as a real-world privacy threat," in *ESORICS*, 2010, pp. 215–231.

[26] E. W. Felten and M. A. Schneider, "Timing attacks on web privacy," in *CCS '00: Proceedings of the 7th ACM conference on Computer and communications security*. New York, NY, USA: ACM, 2000, pp. 25–32.

[27] M. Zalewski, "Rapid history extraction through non-destructive cache timing (v8)," http://lcamtuf.coredump.cx/cachetime/, 2011, [Online; accessed 25-Jan-2012].

[28] L. D. Baron, "Preventing attacks on a user's history through css :visited selectors," http://dbaron.org/mozilla/visited-privacy, 2010.

[29] Iaoss, "NoScript Firefox extension."

[30] P. D. Ryck, L. Desmet, T. Heyman, F. Piessens, and W. Joosen, "Csfire: Transparent client-side mitigation of malicious cross-domain requests," in *ESSoS*, 2010, pp. 18–34.

[31] A. A. E. Ahmed and I. Traore, "European standard en50133-1: Alarm systems. access control systems for use in security applications. part 1: System requirements," 2002.