Analyzing Characteristic Host Access Patterns for Re-Identification of Web User Sessions

Dominik Herrmann, Christoph Gerber, Christian Banse, Hannes Federrath
[firstname.lastname]@wiwi.uni-r.de

Research Group for Management of Information Security
Department of Management Information Systems
University of Regensburg, 93040 Regensburg, Germany

Abstract. An attacker, who is able to observe a web user over a long period of time, learns a lot about his interests. It may be difficult to track users with regularly changing IP addresses, though. We show how patterns mined from web traffic can be used to re-identify a majority of users, i. e. link multiple sessions of them. We implement the web user re-identification attack using a Multinomial Naïve Bayes classifier and evaluate it using a real-world dataset from 28 users. Our evaluation setup complies with the limited knowledge of an attacker on a malicious web proxy server, who is only able to observe the host names visited by its users. The results suggest that consecutive sessions can be linked with high probability for session durations from 5 minutes to 48 hours and that user profiles degrade only slowly over time. We also propose basic countermeasures and evaluate their efficacy.

1 Introduction

With the continuing dissemination of the World Wide Web we are increasingly living our lives online. The websites that are retrieved by an individual reflect – at least to some degree – his or her interests, habits and social network. The URL of some pages may even disclose the user's identity. If one is able to observe a substantial portion of the web traffic of a user over some period of time, he will learn many private details about this user. Many users are willing to trust their ISP, who can trivially intercept all traffic from a dial-up account and attribute it to the respective customer. Malicious observers or other third-party service providers are not supposed to be able to compile profiles that contain users' interests together with their identity, though. Third parties that can easily obtain users' web traffic include open proxy servers, free WiFi hotspots as well as single-hop anonymisation services like Anonymizer.com or the recently launched IPREDator.se. As web browsers usually issue a DNS query before the requested page can be retrieved, the providers of public DNS servers such as OpenDNS or the recently launched Google Public DNS[1] are also part of this group.

A malicious observer can group all requests originating from a single source IP address and (assuming exactly one user per address) attribute all of them

[1] See http://www.opendns.com/ and http://code.google.com/speed/public-dns/.

to a (now pseudonymous) single user. Clearly, in this scenario the attacker's capability to track a user over time mainly depends on the lifetime of the user's IP address. While it is straightforward to track users with static IP addresss, *re-identifying* users with dynamically assigned, frequently changing IP addresses is more challenging. The *web user re-identification attack* addresses this challenge.

In this paper we examine to which extent a passive observer can link the web sessions of a given user solely based on a record of his past activities on the web. Recently, privacy concerns have raised interest in such *re-identification* problems [25]. The first stepping stone for long-term tracking attacks of web users is linking two or multiple *surfing sessions* of individuals, which we address in this paper. In the long run we are interested in a realistic threat assessment of such linkage attacks in real-world environments. Note that we do not examine how to recover the true identity of a web user based on their browsing behaviour in this paper, though. Previous work, e. g. an analysis of the AOL search logs, has shown that at least some users tend to disclose their identity via entering uniquely personally identifying information in web forms or search engines [3]. The more sessions of one user an attacker can link, the more he will learn about his interests and personality – and thus the more likely he will be able to uncover the real-world identity of the user.

For the purpose of our evaluation we model a surfing session to consist of the access frequencies of all hosts a user visits in a certain time window. We will use machine learning techniques to link multiple sessions and analyse how effective a malicious observer (or a third party mentioned above) can re-identify web users. Without loss of generality, we will describe the attack from the perspective of a malicious web proxy server.

Contribution Firstly, we demonstrate that Internet users exhibit characteristic web browsing behaviour that can be exploited for linkage attacks. Our evaluation on a privacy-preservingly collected real-world dataset demonstrates that even an attacker with limited power can exploit characteristic behaviour to re-identify a majority of users on a session-to-session basis. Contrary to previous work, i. e. re-identifying users in 802.11 networks [28], which relies on numerous properties of network traffic, our attack solely utilises destination host access frequencies. Another novelty of our work is the transformation of the raw access frequency vectors to counter the effects of the power-law distribution on access frequencies and a thorough evaluation taking into account the attacker's viewpoint. While previous work operated on monthly traffic aggregates [18] and destination IP addresses, we evaluate our approach for shorter sessions (between 5 minutes and 48 hours) and only rely on host (DNS) names. Furthermore, we discuss and evaluate countermeasures that degrade the effectivity of the attack.

This paper is structured as follows: After reviewing related work in Section 2, we briefly present the data mining techniques used for our attack in Section 3. We continue with our data acquisition methodology in Section 4 before we describe our evaluation methodology and results in Section 5. We present countermeasures in Section 6 and discuss the results in Section 7 before concluding the paper in Section 8.

2 Related work

Closely related to our work are Kumpost's publications [16,17,18], which describe a large-scale study on NetFlow traffic logs. His ultimate goal and approach is quite similar to ours: finding out whether it is possible to pinpoint individual users among others due to their characteristic behaviour in the past. He devises a classifier that compares behavioral vectors of users with a similarity measure based on cosine similarity and shows that inverse document frequencies (IDF) can improve re-identification accuracy. His study differs from ours in several ways, though: Kumpost operates on monthly aggregates of the access frequencies of hosts; on the contrary, we adopt an attacker's point of view and track users on a smaller scale and for shorter timeframes. Furthermore, while Kumpost operates on network traces, we work with a pseudonymized web proxy dataset specifically collected for this purpose. Finally, Kumpost only describes the actual attack, whereas we also discuss and evaluate countermeasures.

Yang's publications [27,36] and especially [35] are also related to our study. Yang studies to which extent samples of 2 to 100 web users can be re-identified with profiling and classification methods from a dataset containing 50,000 users in total. As Yang's focus is the utility of web user profiles for fraud detection and other applications in e-commerce, she does not tackle the problem from our *attacker's view*. To some degree her methodology is comparable to our *simulations*, but there are some differences, which are of relevance for our purpose. For instance, while we concentrate on training sets of size 1, her evaluation focuses on the improvements obtained by the use of multiple labelled training instances (up to 100), which are usually difficult to obtain for the type of attacker we have in mind. Another difference stems from the selection of training and test instances: while Yang selects training and test instances with an arbitrary temporal offset, we explicitly evaluate the influence of the temporal offset between them in order to analyse profile degradation over time.

Also related is the work of Pang et al., which studies an attacker who aims to re-identify users in 802.11 wireless networks [28]. Pang considers a number of properties of network traffic to link multiple sessions of users – even if ephemeral, pseudonymous MAC addresses are used. While they do look at exploiting destination addresses for their linkability attack, their focus lies on characteristics of 802.11 devices such as SSID probes, the size of broadcast packets and MAC protocol fields. Their methodology, which relies on the Jaccard index and a Naïve Bayes classifier with Gaussian kernel density estimation, differs from ours considerably, though.

Data mining techniques have been applied to attack users' privacy in many related user re-identification and de-anonymization studies ([31,15,20,5,23] and most recently [11,34]) and for attacks on anonymized traffic logs [29,9,8]. Web usage mining (cf. [30,6,14,24]) is also a related area of work.

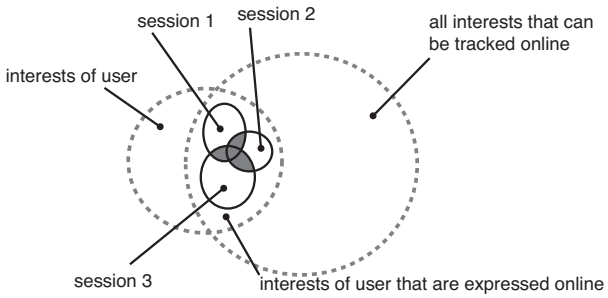


Fig. 1: Venn diagram representation of the web user re-identification problem

3 Re-Identification Methodology

In the following sections we will present our methodology. We assume that most users exhibit at least some part of their interests online, and that they are reflected by the websites they access in a particular surfing session (cf. Fig. 1). Our re-identification attack works on the intersection of two or more sessions of a user. In an ideal world the intersections of one user will not substantially overlap with the intersections of other users as they have a differing set of interests.

Both, term frequencies [37] and host access frequencies [1,4,10], have shown to obey Zipf's law: there is a small number of attributes (terms or hosts) that is part of almost all instances (documents or sessions) and always occurs in large frequencies. As a consequence we conjecture that text mining and web user re-identification can be tackled with similar techniques. Therefore, we model instances using the *vector space model* [2,21,33]. For our analysis, we apply the Multinomial Naïve Bayes classifier, an off-the-shelf text mining technique, and various transformations, which have proven effective for text mining problems, to the input data.

3.1 Modelling the Web User Re-Identification Problem

Our analysis relies on a basic model that captures users' surfing habits. With this model we can reduce the web user re-identification attack to a data mining *classification problem* [33], which can be tackled with various supervised learning methods. We consider each *surfing session* of a user to be an *instance* of a *class* $c_i \in C$, i. e. each class represents all surfing sessions of a specific user. Each instance consists of the web browsing requests sent by a user's machine during one surfing session. From each HTTP request we only use the destination host name (e. g. *www.google.com*).[2] We disregard port, path, filename and other features. Instead of a binary encoding of the fact whether some *host* (e. g. *www.google.com*) has been accessed or not, we take into account the *number of*

[2] Accesses to various sub-domains are not merged, i. e. *www.site.com*, *site.com* and *www1.site.com* are treated as different hosts.

requests to each host within a session to model usage intensity. The order of requests as well as timing information is neglected in our basic model, as we do not expect the behaviour of most users to show significant patterns in those dimensions. There are certainly more sophisticated models conceivable, which may take into account such characteristics.

Note that previous studies [16,17,18] have not relied on host names, but on IP adddresses. While it is certainly possible to carry out the attack with IP addresses only, we deem IP addresses not as suitable as host names: firstly, the IP address of a web server may be subject to frequent changes, secondly, some web sites may use multiple IPs for load distribution and thirdly, virtual hosts may serve multiple different web sites from the same IP address. The instances will reflect user interests more closely, if destination host names are used instead of IP addresses. This is straightforward for the kind of attacker we have in mind, i. e. the provider of a HTTP proxy.

Each instance consists of a multiset $(x_1^{f_{x_1}}, x_2^{f_{x_2}}, \ldots, x_m^{f_{x_m}})$ containing all the hosts x_j and their respective access frequencies $f_{x_j} \in \mathbb{N}_0$ for a given user and session. From the multisets, we obtain *attribute vectors* $\boldsymbol{x} = \mathbf{f} = (f_{x_1}, f_{x_2}, \ldots, f_{x_m})$ for all visited hosts m that are present in the dataset. Even for rather small user groups, those vectors become very sparse as the number of distinct websites increases rapidly.

The re-identification attack consists of two stages. Firstly, the attacker has to obtain a set of k *training instances* $I_{\text{train}} = \{(\boldsymbol{x}_1, c_1), \ldots, (\boldsymbol{x}_k, c_n)\}$; $c_i \in C$; $k \geq n$; $n \leq |C|$ that he labels with class information.[3] Afterwards, he will use a *classifier* to predict the class, i. e. the user, of a number of *test instances* in order to establish a mapping between the sessions contained in the test instances and the sessions within the training instances.

3.2 Multinomial Naïve Bayes (MNB)

The Multinomial Naïve Bayes (MNB) classifier is a well known method for text mining tasks [33]. The choice of the MNB classifier is motivated by the fact that attributes in natural language models and in our model, which relies on host access frequencies, both are distributed according to a power-law, i. e. their frequency distribution is heavy-tailed.

Although Naïve Bayes and related probabilistic classifiers naïvely assume independence of attributes (which is often not the case for real-world problems), they have been applied to many privacy-related classification attacks with great success. Of particular interest for our analysis is the application to traffic analysis problems (cf. [12,38,22,32]) and to website fingerprinting [19,13] in previous

[3] The *class labels* may either be actual real names of the users, in case the attacker already knows them for the training instances or can deduce them using context information. Alternatively, the attacker can use arbitrarily chosen user IDs, i. e. pseudonyms, in case he does not know the real identities of the respective users during the training stage yet. Later on he can substitute the pseudonyms with real-world identifiers, once users have revealed (parts of) their identity by their online activities (which is not within the scope of this paper).

works. We apply the MNB classifier to the host access frequencies within individual user sessions. Given m unique hosts, the classifier evaluates the probability that a given instance $\mathbf{f}$ belongs to some class c_i as:

$$P(\mathbf{f}|c_i) \sim \prod_{j=1}^{m} P(X = x_j|c_i)^{f_{x_j}}$$

The resulting probability is proportional to the product of $P(X = x_j|c_i)$, which is the probability that a certain host x_j is drawn from the aggregated multiset of all host accesses of the training instances of class c_i. The individual probabilities contribute f_{x_j} times to the result, where f_{x_j} is the number of accesses to host x_j in the test instance at hand. In other words: the more often the dominant hosts of the test instance $\mathbf{f}$ appear in the training instances of class c_i, the more likely does instance $\mathbf{f}$ belong to class c_i. The classifier will select the class c_i for with the highest value $P(\mathbf{f}|c_i)$ is observed. For a more formal coverage of the MNB classifier refer to a recent text book by Manning et al. [21].

3.3 Vector Transformations

There are several transformations that – if applied to the raw attribute vectors – have shown to improve the accuracy of classifiers on text mining problems. We will analyse to what extent web user re-identification attacks benefit from them.

TF Transformation Extremely high frequencies of a small number of attributes can overshadow the contribution of the remaining features, which makes it difficult for the classifier to distinguish between instances of different classes. A frequently mentioned solution is to apply a sublinear transformation to the raw occurence frequencies: $\mathrm{f}^*_{x_j} = \log(1 + \mathrm{f}_{x_j})$, the so-called *term frequency (TF) transformation* (cf. [33] for details).

IDF Transformation Using raw vectors all attributes (host frequencies) contribute equally to the resulting vector, regardless of their *relevance*. Popular hosts that are part of a vast majority of instances do not confer much information about a class, though. This problem can be alleviated using the *inverse document frequency (IDF) transformation*: given n training instances the occurrence frequencies f_{x_j} are transformed using the *document frequency* df_x, i. e. the number of instances that contain term x: $f^*_{x_j} = f_{x_j} \cdot \log \frac{n}{\mathrm{df}_{x_j}}$. The application of both of the aforementioned transformations is referred to as *TF-IDF transformation* [33].

Cosine Normalisation (N) Results from empirical research have shown that the accuracy of many classifiers and information retrieval algorithms can be greatly improved by normalizing the lengths of all instance vectors [21, p. 128]. This is usually achieved by applying cosine normalisation, i. e. all frequencies are divided by the Euclidean length of the raw vector: $f^{\mathrm{norm}}_{x_j} = \frac{f^*_{x_j}}{\left\|(f^*_{x_1}, \ldots, f^*_{x_m})\right\|}$. While it stands to reason that cosine normalization is reasonable for text documents, its utility for the web user re-identification problem may seem counterintuitive at first sight: the total number of requests of a session seems to be a promising feature for differentiation, after all.

Table 1: Properties of our proxy user linkability dataset

Duration in days	57
Number of HTTP requests	2,684,736
Number of unique destination hosts	25,124
Transmitted data volume in GiB	110.74

4 Data Acquisition

In this section we will outline our data acquisition methodology and present the dataset used for the evaluation of the user re-identification attack. To collect web surfing data, we recorded the web traffic of 28 web users at the university of Regensburg (cf. Table 1 for descriptive statistics of the dataset).

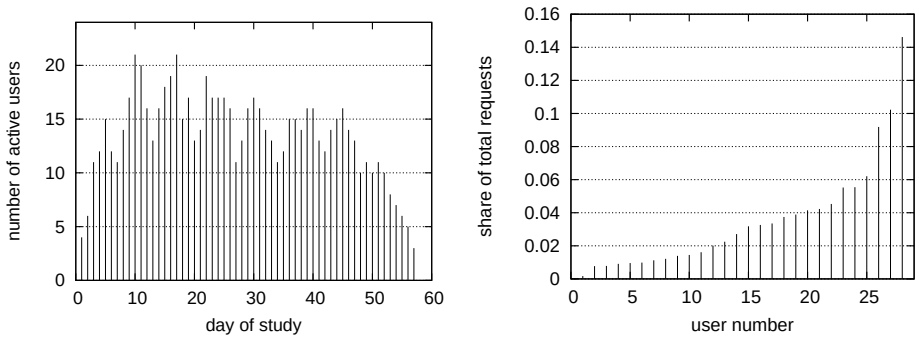
Our participants installed a proxy server (a slightly modified version of Privoxy[4]), which recorded all of their HTTP traffic, on their local client machines. We provided a convenient obfuscation and submission tool that enabled users to anonymize log files on their machines before uploading them to a central server for later collection. The tool labelled the logs with a static user-specific pseudonym (e. g. *RQFSPJ75*) and obscured the requested URLs (see below). To conceal the IP addresses of our participants, we made sure that the log files themselves did not contain any source IP addresses and encouraged the participants to upload their logs using an anonymization service like JonDonym or Tor.[5]

The requested URLs were split into multiple components (scheme, host, port, path) before hostnames and paths were obfuscated using a salted hash-function. The salt value was hard-wired in the obfuscation tool and ensured that there would be a consistent mapping between host names and hash values for all users. The hash function was repeatedly applied to discourage dictionary attacks during the study. Once the study was completed we deleted all references to the salt in order to reduce the risk of dictionary attacks in the future. Our participants were satisfied with the basic level of protection offered by our URL pseudonymization scheme. Even the technically savvy ones, who were familiar with signature and fingerprinting attacks on such log files (cf. [7,15]), were willing to accept the remaining risks.

While our user group is rather small and certainly biased to some degree, our user profiles also have some advantage in comparison to profiles compiled from passively collected flow traces of a large network segment like used by Kumpost [17]: Firstly, the user group is quite homogeneous and shares common interests (24 out of 28 participants are undergraduate or postgraduate students with high affinity towards information technology); this may also be the case in reality for users who share the same proxy server. Secondly, we know as a ground truth that all HTTP requests submitted by a user (i. e. labelled with his pseudonym)

[4] Available for download at http://www.privoxy.org/

[5] Available for download at http://www.jondonym.com/ and http://torproject.org/.



(a) User's participation over time (b) Share of total requests by user

Fig. 2: Overview of dataset used for evaluation

originate from exactly one individual[6], while passively collected profiles may be subject to unobservable influences such as multiple people sharing one IP address or sudden changes in the IP address assignment of a user. Finally, our user profiles are mostly comprised of requests issued via the user's web browser: only 14 % of the participants chose to submit *all* HTTP requests of their machines.

Fig. 2 shows the participation of users over the course of the study as well as the amount of HTTP requests contributed by the individual users. The number of users contributing data on a given day varies between 3 and 22. 15 users or more contributed on at least 50 % of the days, with contributions ranging from 10 to 57 days. The number of HTTP requests submitted by the individual users varies considerably: the 50 % most active users contribute 80 % of the requests, with the most active user contributing 14.6 % of the total number of requests. Note that we will disregard any context information as well as timing information and activity patterns regarding to specific weekdays in this paper. Instead, we will only consider the access frequencies of the destination servers to assess the effectiveness of our web user re-identification attack.

5 Evaluation Methodology and Results

In this section we evaluate the user re-identification attack using our dataset. The evaluation consists of two parts, which allow us to analyse different aspects. In the first part, which we call the *attacker's view*, we merge all the log files of our participants (maintaining the exact timing of the requests). This allows us to evaluate the feasibility of the user re-identification attack if all participants had used a malicious *central proxy server* for our study. The second part consists of *simulations* which analyse the impact of various parameters on the effectiveness of the attack using random samples of our dataset.

[6] This claim is substantiated by the results of an anonymous questionnaire we requested the participants to fill out.

Table 2: Evaluation of predictions from an attacker's point of view for user u (with class c_u) with training instances $x_\cdot^t$ and test instances $x_\cdot^{t+1}$. TP/FP/TN/FN conditions are shown for clarity; cases are sorted according to their evaluation: *correct* $(1, 2)$, *wrong-detectable* $(3, 4)$ and *wrong-undetectable* $(5, 6)$.

(1)	TP:1 FP:0 TN:0 FN:0	u contributed on days t and $t+1$; user's instance x_u was correctly assigned to c_u; attacker can track u
(2)	TP:0 FP:0 TN:1 FN:0	u contributed on day t only; no instance was incorrectly assigned to c_u
(3)	TP:0 FP>1 TN:0 FN:1	u contributed on both days, but x_u^{t+1} was not assigned to c_u; multiple instances $x_v; v \neq u$ were assigned to c_u
(4)	TP:1 FP>0 TN:0 FN:0	u contributed on both days; x_u^{t+1} was assigned to c_u; at least one instance $x_v; v \neq u$ was assigned to c_u
(5)	TP:0 FP:0 TN:0 FN:1	u contributed on both days; but no instance was assigned to c_u at all; attacker believes there is no x_u^{t+1} and loses track of u; attacker confuses this prediction with (2)
(6)	TP:0 FP:1 TN:0 FN:0	u contributed on both days, but x_u^{t+1} was not assigned to c_u; one instance $x_v; v \neq u$ was assigned to c_u; attacker confuses v with u; attacker confuses this prediction with (1)

5.1 Attacker's View

We start out with an attacker on a proxy server who exploits characteristic surfing patterns to re-identify individual users on consecutive days, i. e. we consider sessions with a duration of 24 hours (we study other session times and non-consecutive sessions in Section 5.2).

Therefore, we assume that on one day t the attacker decides to track a specific user u^t from now on (e. g. due to a intriguing request of that user). The attacker chooses u from the set of all users U^t who are present on day t. For the attack he sets up a classifier with $|U^t|$ classes c (one for each user), and trains the classifier with the available instances $x_\cdot^t$ of all users from U^t (one instance per user). On the next day, the attacker tries to find the instance x_u^{t+1}, i. e. all the instances that are predicted to belong to class c_u are of interest. Ideally, only the correct instance x_u^{t+1} will be assigned to c_u.

Due to the peculiarities of the attacker's view there are more than the four canonical evaluation results (true positives, false positives, true negatives and false negatives) [33]. Table 2 contains an overview of our more differentiated evaluation scheme. The prediction of the classifier can either be *correct* $(1, 2)$, *wrong-detectable* $(3, 4)$ or *wrong-undetectable* $(5, 6)$.

Evaluation Results We iterate over all days and users and evaluate the prediction of the MNB classifier for the transformations presented in Section 3.3. Each

Table 3: Classification accuracy for attacker's view (AV) and simulation (SIM), i. e. the proportion of user sessions for which the classifier correctly and unambiguously predicted the correct class (1) or correctly predicted that the user did not participate on the second day (2).

	none	N	IDF	IDFN	TF	TFN	TFIDF	TFIDFN
(AV)	60.5 %	62.9 %	65.0 %	62.8 %	56.0 %	**73.1 %**	66.1 %	72.8 %
(SIM)	55.5 %	56.2 %	65.0 %	60.2 %	53.3 %	77.1 %	68.5 %	**80.1 %**

prediction is evaluated independently, i. e. the conceived attacker is stateless and does not change his behaviour based on the predictions on previous days. For each experiment we report the overall *classification accuracy*, i. e. the proportion of correct predictions $(1, 2)$. An overall comparison of the various transformations is shown in the (AV) row in Table 3. Cosine normalisation (N) increases the accuracy of the classifier significantly when applied in combination with one of the other transformations. The TFN transformation leads to the highest number of correct predictions: 73.1% of all day-to-day links were correctly established, i. e. user u was either re-identified unambiguously (1) or the classifier correctly reported that u was not present on day $t + 1$ any more (2). Note that the utility of the IDF transformation is rather limited in the attacker's view scenario. This counterintuitive finding can be explained by the relatively small number of only 765 predictions in the attacker's view scenario.

While already this basic attack achieves respectable results, there is certainly room for improvements. We present only one of them here: *learning*. We have found that the accuracy of the classifier can be increased considerably, if the attacker is not stateless, but is allowed to "learn", i. e. he can add already predicted instances x_u^{t+1} to the set of training instances for user u, if the prediction *appears* to be correct (1,6). In the case of the MNB classifier and the TFN transformation, the proportion of *correct* decisions (accuracy) increased from 73.1 % to 77.6 %, the proportion of *detectable errors* decreased from 14.5% to 12.5 % and the proportion of *undetectable errors* decreased from 12.4 % to 9.8 %.

5.2 Simulations

The results obtained from the *attacker's view* experiments indicate that a central proxy can carry out the web user re-identification attack for small user groups like ours. Due to its dynamic nature, i. e. not all users having participated on all days (cf. Fig. 2a), the *attacker's view* is not very suitable for analysing influence factors that determine the effectiveness of the attack, though. Thus, we will resort to simulations, in which we set up well-defined and balanced scenarios, to gain further insights.

Each simulation experiment works on a random sample of training and test instances drawn from the whole dataset. For each user 10 pairs of training and

testing sessions are drawn for each experiment (iterations). The properties of the pairs are controlled by a number of parameters. Only one parameter is varied in each experiment to analyse its influence. The varied parameters are:

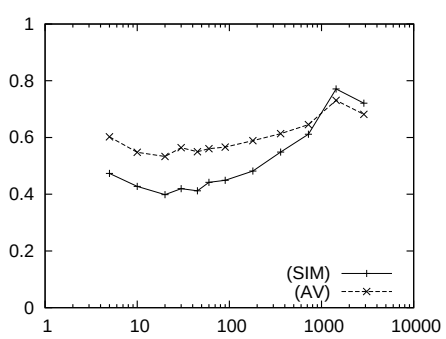
- session duration in minutes (default: 1440, i. e. 1 day),
- number of simultaneous users (default: 28),
- offset between the last training session and the test session (default: the session duration, i. e. adjacent sessions) and
- number of consecutive training instances (default: 1).

The default setup simulates all 28 users concurrently surfing on 10 days (iterations), i. e. for each iteration there are 28 training sessions (1 for each user), each one capturing all requests of the user within one day. Training and test sessions are *not* drawn independently, though: for each user the random session selection process prefers training sessions, which have a (chronologically) immediately succeeding session for the respective user. The succeeding session will then be selected as the test session. This ensures that the parameter "offset between the last training and the test session" equals the session duration for all users. The classifier will be trained with the training sessions (which may in fact come from different days in our real-world dataset) and will have to make a prediction for each of the 28 test sessions. This training and prediction will be repeated for the 10 randomly drawn session pairs (iterations).

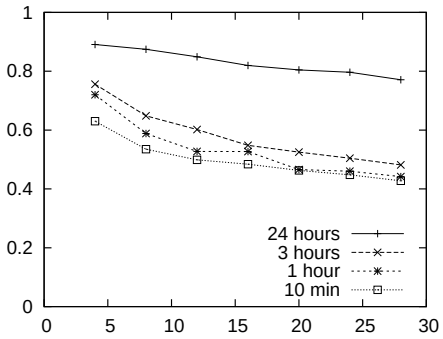
The simulation results are obtained by repeating each experiment 25 times and taking the average of the obtained accuracy. This approach incorporates a large proportion of the dataset in each experiment: the classifier makes $25 \cdot 10 \cdot 28 = 7000$ predictions per simulation experiment. The results for the application of the various text mining transformations are shown in the (SIM) row of Table 3. The TFIDFN transformation achieves slightly better results than the TFN transformation here. The results of all of the following simulations were obtained using the TFN transformation, though, which has lower computational costs and still offers comparable accuracy.

Evaluation Results The results of the simulations are summarized in Fig. 3 for various session durations. Fig. 3a shows that the accuracy of the classifier decreases once session durations become shorter than one day (1440 minutes), which we found is due to the smaller amount of distinct sites and issued requests visited within them. Thus, the information amount available to the classifier decreases. The accuracy increases once again for short sessions below 30 minutes. This is partly due to users' activites spanning session boundaries, which increases the linkability of two adjacent sessions. Furthermore, accuracy increases with decreasing numbers of concurrent users (Fig. 3b), which explains the higher accuracy of the classifier for the attacker's view scenario.

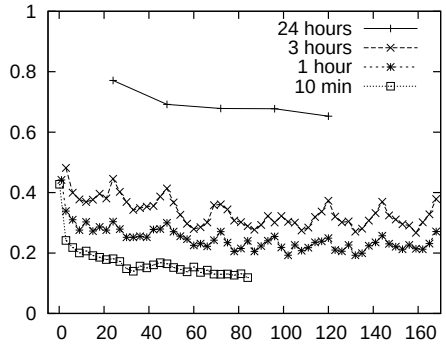
Fig. 3c shows that the quality of the user profiles deteriorates only moderately over time. The waveform patterns in the plot for session durations of 1 and 3 hours have a periodicity of 24 hours. Thus, it is easier to link two sessions of a user if they are obtained at the same *time of day* on different days. Apparently, our users exhibit different behaviour at different times during the day.



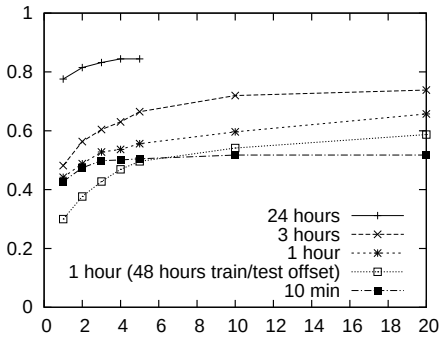
(a) Duration of a user's session in minutes



(b) Number of concurrent users



(c) Training/test instance offset (hours)



(d) Number of training instances

Fig. 3: Simulation results: influence of various parameters on proportion of correctly classified sessions (y-axis);

According to Fig. 3d the accuracy will increase if the attacker manages to obtain not only 1 but multiple consecutive training sessions of a given user or is able to correctly link multiple consecutive sessions of a user (cf. "learning" in Section 5.1). While the gain in accuracy caused by an additional training instance diminishes quite fast for immediately adjacent sessions, multiple training instances can be very useful when it comes to test instances whose offset to the training instance is larger. This becomes evident in Fig. 3d by comparing the slopes of the two curves supplied for 1-hour sessions with training/test offsets of 1 hour and 48 hours. For the latter the accuracy increases more rapidly for up to 5 additional training instances.

5.3 Linkability Metric

We analyzed the dataset for peculiarities that can explain the effectiveness of the classifier. Therefore, we constructed a numerical *host linkability* metric $L \in [0; 1]$ that captures the degree of re-identifiability of a user u that is caused by accessing

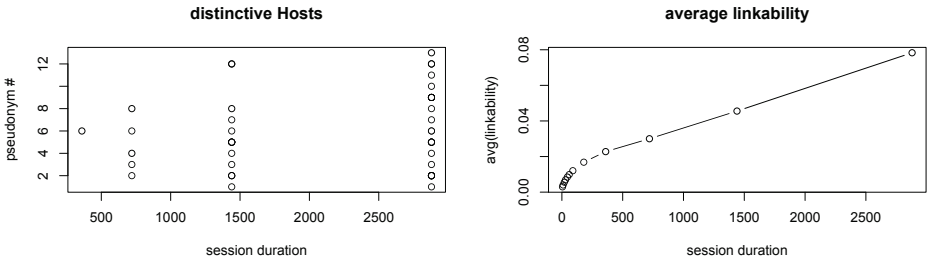


Fig. 4: Distinctive hosts and average linkability for various session durations

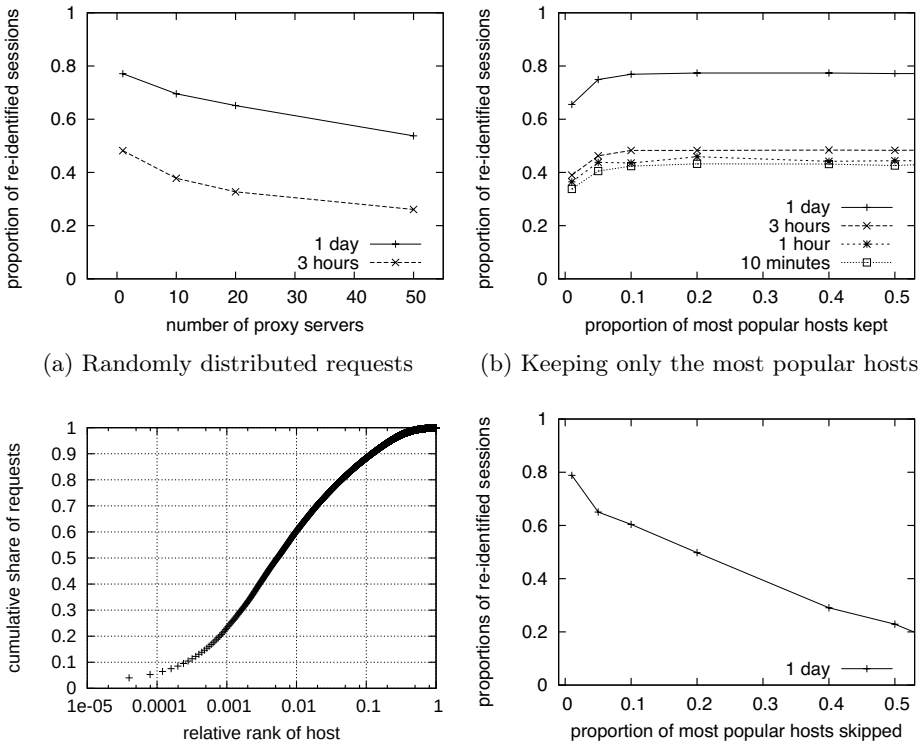
a specific host h. With this metric we can uncover hosts that almost immediately identify users once they request a website from them (cf. also [17, p. 62] for a different approach). Considering the multiset of website requests R^u of user u, the multiset of website requests R_h involving host h, the set of sessions S^u of user u and the set of sessions S_h involving host h, we obtain the host linkability as follows:

$$L(h,u) = \frac{|R^u \cap R_h|}{|R_h|} \cdot \frac{|S^u \cap S_h|}{|S^u|}$$

In words: A host h allows for immediate re-identification of a user u, if h is only accessed by u and if u visits h in each of his sessions; this is expressed by a host linkability value of $L(h,u) = 1$. If an attacker knew the hosts with $L = 1$ – we call them *distinctive hosts* – he wouldn't have to rely on our classification technique but could directly re-identify the respective users. We found 17 distinctive hosts in our dataset for nine users with a session time of 1440 minutes (cf. Fig 4). If a host is *distinctive* for a user for a session duration d_a, it will also be distinctive for this user for all session durations $d_b > d_a$. With decreasing session duration the linkability values for all hosts are decreasing as well, i. e. it is less likely to encounter a distinctive host in shorter sessions.

6 Countermeasures

A user can blur his behavioral profile by distributing his web requests over multiple (non-colluding) proxy servers (similar to the ideas proposed by Olivier [26]). A single server will then only see a subset of the requests of the user. There are many conceivable variants of such a distribution scheme: e. g. based on time (switching the server at regular intervals) or based on destination (all requests for hosts (h_1, h_2, h_3) are sent to server s_1, requests for hosts (h_4, h_5, h_6) are sent to a different server s_2. We leave the design and evaluation of various strategies open for future work. Instead we only analyse a basic strategy, which may serve as a baseline for benchmarking: randomly distributing all the requests of a user over multiple proxies. According to the results for 1, 10, 20 and 50 servers (cf. Fig. 5a), this strategy is effective, but not very efficient.



(a) Randomly distributed requests

(b) Keeping only the most popular hosts

(c) Host ranking according to requests

(d) Skipping the most popular hosts

Fig. 5: Effectiveness of countermeasures

Removing requests from log files In principle the web user re-identification attack is also applicable to log files of proxy servers that are shared or publicly available. There is a number of obfuscation tools that help to protect the privacy of the users whose requests are contained in the logs. Some tools (such as *tcpmkpub*, *tcpdpriv* or the Perl module *NetAddr::IP::Obfuscate*) rely on a consistent hashing or obfuscating scheme of IP addresses, which ensure that a given input address is always mapped to the same obfuscated output address. Thus, our web user re-identification attack can be applied to track users with dynamic IP addresses in such log files without modification. It could also be applied, if the proxy operator changed the mapping scheme from time to time.

In order to counter the attack a proxy operator may come to the conclusion to only share requests to the most popular hosts, which are not supposed to convey any personally identifying information. To evaluate the validity of this assumption we have repeated the simulation, restricting it to the most popular 1 %, 5 %, 10 %, 20 %, 40 % and 50 % of the hosts according to a descending ranking of hosts, which is based on the total number of requests they attracted

(see Fig. 5c). The somewhat surprising results indicate that this approach cannot prevent the user re-identification attack: the classifier can still link more than 65 % of the 1-day sessions (cf. Fig. 5b) if instances are only based on the 1 % (= 251) most popular hosts. Due to the long-tailed distribution of access frequencies the log files contain only 60 % of the total requests in this case (cf. Fig. 5c). The utility of this countermeasure degrades fast: if 10 % or more of the most popular hosts are kept in the log file, accuracy values will not be affected significantly any more. Classification accuracy is also only moderately affected if the most popular hosts are *skipped* (cf. Fig. 5d): training the classifier on 1-day sessions after having removed the 50 % most popular hosts (which is equivalent to skipping 99.91 % of all requests!) still results in an accuracy of 22.9 %. Whether log files that have been stripped in such a way are of any practical use any more, certainly depends on the particular application at hand.

Anonymization services Instead of distributing their requests over multiple proxy servers, users can also rely on anonymisation services like Tor or JonDonym. These services prohibit eavesdropping by local adversaries and consequently also protect against any re-identification attacks carried out by them. The use of anonymisation services may also introduce new risks, though: in mix networks the exit node learns the true destination hosts as requested by its users. The Tor network uses circuits, which relay a user's traffic over a single exit router. After 10 minutes a circuit is abandoned and a new circuit with another exit node is set up. If a Tor user relayed all web requests over a single exit node (which is the default as of now), the exit node could apply our methodology to construct user session and create a MNB classifier to re-identify users. Collaborating exit nodes could share such profiles to track users across multiple exit nodes, which would seriously degrade their privacy. The attack could be prevented, if the Tor client routed web traffic over multiple exit nodes concurrently.

7 Discussion

Due to the limited scope of our study, we cannot precisely assess the real threat of user re-identification on the web. The small number of users may limit the generalisability of our results, but not of our methodology. We are already in the process of applying it to large DNS log files with several thousand users. Even for this different, more difficult problem our first results are promising: we are able to re-identify up to 50 % of the users about 80 % of the time.

For the purpose of evaluation we modelled a *user session* as a rigid time span (e. g. 10 minutes or 24 hours). As a matter of fact, our evaluation tools will erroneously distribute contiguous requests across two sessions, if the true user session crosses our session boundaries, which decreases the difficulty of the classification problem – at least for immediately adjacent sessions. This bias could be cured with a more realistic session splitting method, e. g. by taking into account the results of [6,30], which empirically derive actual session boundaries.

The presented basic form of the attack can not only be carried out by proxy servers or DNS servers, but by any eavesdropper in general. We are aware of the fact that we disregard promising pieces of information, which may be available to some attackers, such as the whole URL or request timing. Web proxies could also inspect the contents of the HTTP messages for identifying information, e. g. usernames and street addresses. While our methodology can be extended to support such attributes, we believe that in reality the biggest improvements will stem from the inclusion of context knowledge: a user who just received his driver's license might visit many hosts like *myfirstcar.com* and *firstcar.com* over a period of several days. Future work might take this into account by employing a semantic model to group hosts according to activities or actual interests.

Finally, we want to point out that our scenario in mind, a closed user group using a single proxy server, allows us to make a closed-world assumption, i. e. each instance belongs to a user of that group. Consequently, our classifier will output a prediction for each and every test instance no matter how likely it is. In some real-world situations, e. g. tracking one user among thousands of unknown users, this approach will cause a false alarm for the majority of instances. Adapting the methodology to cope with such scenarios (e. g. by using a probability threshold or reject class) is certainly an interesting area for future work.

8 Conclusion

Using a privacy-preservingly collected real-world dataset we have demonstrated that an adversary can re-identify web users based on their past browsing behaviour. Thus, malicious providers of (small-scale) web proxies may be able to track their users. Profile degradation over time is only moderate and it can be alleviated using multiple training instances. According to our results, countermeasures such as distributing web requests over multiple proxies can reduce the accuracy of our attack, but they come at a considerable cost.

Our technique is based on the observed access frequencies of hosts. It does not depend on any timing information or context knowledge, and it is totally agnostic of the type of host or the actual contents retrieved. Instead, we exploit the diversity on the World Wide Web: specific user interests (such as reading news or social networking) can be satisfied at a large number of different sites, which is reflected by the long-tailed distribution of access frequencies. Consequently, web user re-identification may even succeed for users with very similar interests – as long as they have a distinct preference regarding the websites where they pursue them.

Acknowledgments

This work has been partially sponsored and supported by the European Union EFRE project. The authors are grateful to the participants of the study who contributed their web requests. This paper has benefitted from fruitful discussions with Jacob Appelbaum, Karl-Peter Fuchs, Nico Görnitz, Konrad Rieck, Florian

Scheuer, Rolf Wendolsky, Benedikt Westermann and from helpful comments by the anonymous reviewers.

References

1. Lada Adamic and Bernardo Huberman. Zipf's Law and the Internet. *Glottometrics*, 3(1):143–150, 2002.
2. Ricardo Baeza-Yates and Berthier Ribeiro-Neto. Modern information retrieval. *New York, Addision Wesley*, 1999.
3. Michael Barbaro and Tom Zeller. A Face is Exposed for AOL Searcher No. 4417749. *The New York Times*, August 9, 2006.
4. Lee Breslau, Pei Cue, Pei Cao, Li Fan, Graham Phillips, and Scott Shenker. Web Caching and Zipf-like Distributions: Evidence and Implications. In *In INFOCOM*, pages 126–134, 1999.
5. Justin Brickell and Vitaly Shmatikov. The cost of privacy: destruction of data-mining utility in anonymized data publishing. In *KDD '08: Proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 70–78, New York, NY, USA, 2008. ACM.
6. Lara D. Catledge and James Edward Pitkow. Characterizing Browsing Behaviors on the World-Wide Web. *Georgia Institute of Technology*, 1995.
7. Scott E. Coull, Michael P. Collins, Charles V. Wright, Fabian Monrose, and Michael K. Reiter. On Web Browsing Privacy in Anonymized NetFlows. In *Proceedings of the 16th USENIX Security Symposium*, Boston, MA, August 2007.
8. Scott E. Coull, Charles V. Wright, Angelos D. Keromytisz, Fabian Monrose, and Michael K. Reiter. Taming the devil: Techniques for evaluating anonymized network data. In *Proceedings of the 15th Network and Distributed Systems Security Symposium*, 2008.
9. Scott E. Coull, Charles V. Wright, Fabian Monrose, Michael P. Collins, and Michael K. Reiter. Playing devil's advocate: Inferring sensitive information from anonymized network traces. In *in Proceedings of the Network and Distributed System Security Symposium*, pages 35–47, 2007.
10. Mark E. Crovella and Azer Bestavros. Self-similarity in World Wide Web traffic: evidence and possible causes. *IEEE/ACM Trans. Netw.*, 5(6):835–846, 1997.
11. Peter Eckersley. How Unique Is Your Web Browser? Technical report, Electronig Frontier Foundation, 2009.
12. Jeffrey Erman, Anirban Mahanti, and Martin Arlitt. Internet Traffic Identification using Machine Learning. In *Proceedings of IEEE Global Telecommunications Conference (GLOBECOM)*, pages 1–6, San Francisco, CA, USA, November 2006.
13. Dominik Herrmann, Rolf Wendolsky, and Hannes Federrath. Website fingerprinting: attacking popular privacy enhancing technologies with the multinomial naïve-bayes classifier. In *CCSW '09: Proceedings of the 2009 ACM workshop on Cloud computing security*, pages 31–42, New York, NY, USA, 2009. ACM.
14. Melanie Kellar, Carolyn Watters, and Michael Shepherd. A field study characterizing Web-based information-seeking tasks. *Journal of the American Society for Information Science and Technology*, 58(7):999–1018, 2007.
15. Dimitris Koukis, Spyros Antonatos, and Kostas G. Anagnostakis. On the Privacy Risks of Publishing Anonymized IP Network Traces. In *Communications and Multimedia Security*, pages 22–32, 2006.

16. Marek Kumpošt. Data Preparation for User Profiling from Traffic Log. *Emerging Security Information, Systems, and Technologies, The International Conference on*, 0:89–94, 2007.
17. Marek Kumpošt. *Context Information and user profiling*. PhD thesis, Faculty of Informatics, Masaryk University, Czeck Republic, 2009.
18. Marek Kumpošt and Vašek Matyáš. User Profiling and Re-identification: Case of University-Wide Network Analysis. In *TrustBus '09: Proceedings of the 6th International Conference on Trust, Privacy and Security in Digital Business*, pages 1–10, Berlin, Heidelberg, 2009. Springer-Verlag.
19. Marc Liberatore and Brian Neil Levine. Inferring the Source of Encrypted HTTP Connections. In *CCS '06: Proceedings of the 13th ACM conference on Computer and communications security*, pages 255–263, New York, NY, USA, 2006. ACM Press.
20. Bradley Malin and Edoardo Airoldi. The Effects of Location Access Behavior on Re-identification Risk in a Distributed Environment. In *Privacy Enhancing Technologies*, pages 413–429, 2006.
21. Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to Information Retrieval*. Cambridge University Press, Cambridge, UK, 2008.
22. Andrew W. Moore and Denis Zuev. Internet traffic classification using bayesian analysis techniques. In *SIGMETRICS '05: Proceedings of the 2005 ACM SIGMETRICS international conference on Measurement and modeling of computer systems*, pages 50–60, New York, NY, USA, 2005. ACM Press.
23. Arvind Narayanan and Vitaly Shmatikov. Robust de-anonymization of large sparse datasets. *IEEE Symposium on Security and Privacy*, pages 111–125, 2008.
24. Hartmut Obendorf, Harald Weinreich, Eelco Herder, and Matthias Mayer. Web Page Revisitation Revisited: Implications of a Long-term Click-stream Study of Browser Usage. In *CHI 2007*, pages 597 – 606. ACM, ACM Press, 5 2007.
25. Paul Ohm. Broken Promises of Privacy: Responding to the Surprising Failure of Anonymization. *Social Science Research Network Working Paper Series*, August 2009.
26. Martin S. Olivier. Distributed Proxies for Browsing Privacy: a Simulation of Flocks. In *SAICSIT '05: Proceedings of the 2005 annual research conference of the South African institute of computer scientists and information technologists on IT research in developing countries*, pages 104–112, , Republic of South Africa, 2005. South African Institute for Computer Scientists and Information Technologists.
27. Balaji Padmanabhan and Yinghui Yang. Clickprints on the Web: Are there signatures in Web Browsing Data? *Working Paper Series*, October 2006.
28. Jeffrey Pang, Ben Greenstein, Ramakrishna Gummadi, Srinivasan Seshan, and David Wetherall. 802.11 user fingerprinting. In *MobiCom '07: Proceedings of the 13th annual ACM international conference on Mobile computing and networking*, pages 99–110, New York, NY, USA, 2007. ACM.
29. Ruoming Pang, Mark Allman, Vern Paxson, and Jason Lee. The devil and packet trace anonymization. *SIGCOMM Comput. Commun. Rev.*, 36(1):29–38, 2006.
30. Jaideep Srivastava, Robert Cooley, Mukund Deshpande, and Pang-Ning Tan. Web usage mining: discovery and applications of usage patterns from Web data. *SIGKDD Explor. Newsl.*, 1(2):12–23, 2000.
31. Latanya Sweeney. k-anonymity: A model for protecting privacy. *International Journal of Uncertainty Fuzziness and Knowledge Based Systems*, 10(5):557–570, 2002.

32. Nigel Williams, Sebastian Zander, and Grenville Armitage. A preliminary performance comparison of five machine learning algorithms for practical IP traffic flow classification. *SIGCOMM Comput. Commun. Rev.*, 36(5):5–16, 2006.
33. Ian H. Witten and Eibe Frank. *Data Mining. Practical Machine Learning Tools and Techniques*. Elsevier, San Francisco, 2005.
34. Gilbert Wondracek, Thorsten Holz, Engin Kirda, and Christopher Kruegel. A Practical Attack to De-Anonymize Social Network Users. *iseclab.org*.
35. Yinghui Yang. Web user behavioral profiling for user identification. *Decision Support Systems*, 49:261–271, 2010.
36. Yinghui (Catherine) Yang and Balaji Padmanabhan. Toward user patterns for online security: Observation time and online user identification. *Decision Support Systems*, 48:548–558, 2008.
37. George Kingsley Zipf. *The psycho-biology of language. An introduction to dynamic philology*. M.I.T. Press, Cambridge/Mass., 2nd edition, 1968.
38. Denis Zuev and Andrew W. Moore. Traffic Classification Using a Statistical Approach. In *Passive and Active Network Measurement*, pages 321–324, 2005.