# POSTER: How Dangerous is My Click? Boosting Website Fingerprinting By Considering Sequences of Webpages

Asya Mitseva
Brandenburg Technical University
Cottbus, Germany
asya.mitseva@b-tu.de

Jan Pennekamp
RWTH Aachen University
Aachen, Germany
jan.pk@comsys.rwth-aachen.de

Johannes Lohmöller
RWTH Aachen University
Aachen, Germany
lohmoeller@comsys.rwth-aachen.de

Torsten Ziemann
Brandenburg Technical University
Cottbus, Germany
torsten.ziemann@b-tu.de

Carl Hoerchner
Brandenburg Technical University
Cottbus, Germany
carl.hoerchner@b-tu.de

Klaus Wehrle
RWTH Aachen University
Aachen, Germany
wehrle@comsys.rwth-aachen.de

Andriy Panchenko
Brandenburg Technical University
Cottbus, Germany
andriy.panchenko@b-tu.de

## ABSTRACT

Website fingerprinting (WFP) is a special case of traffic analysis, where a passive attacker infers information about the content of encrypted and anonymized connections by observing patterns of data flows. Although modern WFP attacks pose a serious threat to online privacy of users, including Tor users, they usually aim to detect single pages only. By ignoring the browsing behavior of users, the attacker excludes valuable information: users visit multiple pages of a single website consecutively, e.g., by following links. In this paper, we propose two novel methods that can take advantage of the consecutive visits of multiple pages to detect *websites*. We show that two up to three clicks within a site allow attackers to boost the accuracy by more than 20% and to dramatically increase the threat to users' privacy. We argue that WFP defenses have to consider this new dimension of the attack surface.

## CCS CONCEPTS

**• Security and privacy → Pseudonymity, anonymity and untraceability**; **• Networks → Network privacy and anonymity**.

## KEYWORDS

Traffic Analysis; Website Fingerprinting; Web Privacy

**ACM Reference Format:**
Asya Mitseva, Jan Pennekamp, Johannes Lohmöller, Torsten Ziemann, Carl Hoerchner, Klaus Wehrle, and Andriy Panchenko. 2021. POSTER: How Dangerous is My Click? Boosting Website Fingerprinting By Considering Sequences of Webpages. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security (CCS '21), November 15–19, 2021, Virtual Event, Republic of Korea.* ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/3460120.3485347

## 1 INTRODUCTION

Today, Tor [2] is the most popular low-latency anonymization network used to hide the identity (i.e., IP address) of Internet users and to bypass country-level censorship. To achieve anonymity, Tor users encrypt their data in multiple layers and transmit it through a chain of three volunteer nodes. Thus, Tor promises to hide the relationship between users and their communication partners from a *local passive observer*, e.g., an ISP, located on the link between the Tor user and the first anonymization node [2]. However, Tor leaks information about the number, direction, and timing of transmitted packets, which enables the mounting of sophisticated attacks such as website fingerprinting (WFP) [7–9]. In WFP, the attacker aims to identify the content (i.e., the website visited) of encrypted and anonymized connections by analyzing patterns of communication. He collects traces of multiple page loads for each of his websites of interest, extracts patterns (i.e., *fingerprints*) from the recorded traffic, and applies machine learning (ML) to train a classifier to recognize them. Finally, he uses the trained classifier to detect which website has been visited by observing an unknown trace of a real user. Although modern WFP attacks [4, 8, 9] achieve more than 90% of classification accuracy in laboratory settings, their efficiency in real world is still highly debated due to the use of unrealistic assumptions and the huge universe size of the World Wide Web.

Currently, related work [4, 7–9] mainly focuses on the detection of concrete index pages through isolated page loads, instead of the site a visited page belongs to (the de-facto goal of a real adversary). Only a few works [7, 8] examine a more realistic scenario, in which users can visit both index and non-index pages of different websites. However, these studies do not analyze the danger of WFP when users browse multiple pages of a given website. On the other hand, real users visit several pages of a single site consecutively, e.g., by following links. Hence, if the adversary can exploit the additional information leaked through the set of pages belonging to the same

*CCS '21, November 15–19, 2021, Virtual Event, Republic of Korea*
© 2021 Copyright held by the owner/author(s).
ACM ISBN 978-1-4503-8454-4/21/11.
https://doi.org/10.1145/3460120.3485347

website and visited by a user one after another, we argue that WFP will become vastly more dangerous than previously expected.

In line with this revised evaluation setting, we propose two novel WFP strategies, *voting-based* and *HMM-based*, that consider the set of pages of a single site accessed consecutively by a user. Although Cai et al. [1] have already apllied a Hidden Markov Model (HMM) to model a specific user behavior, the authors used a very limited dataset and analyzed neither the influence of the number of observed pages of a website nor the impact of different user behaviors on the accuracy—the main contributions of our work. By using our WFP strategies, we show that two, at most three, clicks within a website allow to boost the accuracy by more than 20% and brings it into the alarming area. Moreover, our methods improve the attack even without the knowledge about the exact sequence of visited pages, rendering it even more dangerous.

## 2 OUR FINGERPRINTING STRATEGIES

We aim to detect a website by observing a number of pages of that specific site that are visited by a user one after another. These consecutive visits leak information about the classification that can be sourced to *refine* single predictions for individual pages. We analyze two strategies that exploit this leakage: $(i)$ *voting-based* combining the predictions for separate pages of a single site without considering the order of their visits, and $(ii)$ *HMM-based* using the knowledge about the sequence of visited pages to detect the website.

**Voting-based.** We use a classifier that is trained on different websites represented through both their index and non-index pages. For each testing page, the classifier computes a set of probability values associated with the likelihood that the given page load belongs to each of these sites. Next, for each testing set of observed pages, we multiply the probability values of these pages. As a result, for each website class, we obtain a single probability for each testing set of observed pages and the website class with the highest likelihood yields our final prediction.

**HMM-based.** We create a separate HMM model for each website, in which pages correspond to different states and state transition probabilities represent the probability a user would navigate from one page to another. As the majority of websites consists of a large number of pages, the use of a separate state for a single page does not scale. Thus, we use clustering to aggregate several webpages that look similar and have the same link connectivity to other pages of the same website into a single HMM state. As the number of clusters varies for different websites, we use the DBSCAN clustering algorithm [3], which does not require any prior information about the number of clusters to be created. The set of created clusters represent the set of hidden states in our HMM model and we train a separate classifier on each of these clusters.

Beside the set of hidden states, we also need to define the set of observations, the set of transition probabilities indicating the likelihood of generating a given observation upon transitioning to a certain hidden state, the set of initial probabilities, and the set of observation probabilities to complete the HMM model for each website. The set of observations corresponds to the set of predicted cluster labels for each testing page. To derive the set of transition probabilities, we use two sources of data: $(i)$ randomly-generated user browsing sessions describing sequences of pages, and $(ii)$ a sitemap graph of each website containing available pages and the link relationships between them. The set of start probabilities is the relative frequency of clusters (counted for a set of training sessions) containing the first page in a session. The set of observation probabilities describes the confusion between the two sets of predicted labels (observations) and the real labels (hidden states), i.e., how many training pages labeled as class $i$ are predicted as class $j$. To obtain the final prediction for a sequence of pages, we sum the predicted probability values of each page for each website and then multiply the aggregated probabilities of the pages in the sequence. The website class with the highest likelihood is our final prediction.

## 3 DATASET

A typical user browsing session usually contains pages of less popular websites, e.g., local newspapers, small sport clubs [5]. Thus, we compiled a dataset of 100 *websites* that consists of both less popular websites covering different categories, different layouts, and contents from different regions in the world and Alexa Top websites. For each website, we then create a sitemap graph that is used to collect randomly-generated user browsing sessions.

**Generating Sitemap Graphs.** For each website, we created a sitemap graph containing data about available webpages and the link relationship between them. Although some websites offer a hierarchical overview documents of their pages, these documents do not always provide data on page linkability. Thus, we used a different strategy to collect the sitemap graphs. First, for each site, we gather the URLs of its index page and four additional, popular pages of it that were found from Google, i.e., to simulate that users access a website not only through its index page but also through an already known link, using a bookmark or by querying a search engine [5]. Starting from one of these five pages, we then extract all URLs from that page referring to the same website. We group the collected URLs based on their position on the page, i.e., whether they are located in the navigation section or in the footer, and exclude groups of URLs that are typically less visited by users, e.g., privacy policy and legal notice pages. From the remaining groups, we randomly select ten groups of URLs, fetch one random URL from each of these groups to simulate a user click, and repeat the procedure described above to decide on the next click. The crawling of URLs terminates once we reach a depth of ten pages for each website and have gathered at least 2000 unique pages. Finally, we build a directed graph where each node represents a URL and an edge between two nodes corresponds to a link between these URLs. For this graph, we consider all seen URLs regardless whether they were selected by the sampling for further steps or not.

**Generating User Sessions.** Although a stored browser history would be a reliable source of real user sessions, it can reveal confidential data about users and usually is kept private. Thus, we use the gathered sitemap graphs to synthetically create a set of user sessions while ensuring that they exhibit realistic characteristics, as described by Kumar et al. [5]. As users can access a website in different ways, we use either the index page or one of the four additional pages of a site to start a user session. As Miller et al. [6], we execute a random walk over the sitemap graph of that website to sample the rest of the user session, whereas we prefer pages that have been visited neither in the current nor in any previously

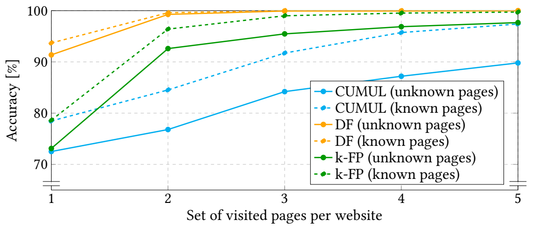

**Figure 1: Accuracy achieved by state-of-the-art WFP attacks.**

generated session. The latter increases the diversity between different sessions (and, thus, complicates the WFP attack). In total, we sampled 10 sessions for each website containing 10 pages per session and at least 50 unique pages per website.

**Collection of Traffic Traces.** We rely on an existing method [8] that operates Tor Browser 7.5.6 to collect 20 traces for each of the pages in our user sessions. Like related work [4, 9], we further reconstruct the corresponding Tor cells exchanged for each page load by applying a previously-used data extraction method [8].

## 4 EVALUATION

Next, we demonstrate the effectiveness of our novel WFP strategies.

**Voting-based.** We first analyze how our voting-based scheme influences the accuracy of different state-of-the-art WFP attacks for different sets of testing pages per website in a closed-world scenario (i.e., the attacker knows the set of all visited websites). We consider two evaluation scenarios: $(i)$ the adversary knows all webpages of a given website that can be visited by a user, and $(ii)$ the attacker can use only a subset of the available pages belonging to a given website for training. Based on these scenarios, we apply a 10-fold cross-validation (CV) either with respect to the number of available traces per webpage or with respect to the number of available pages per website. Figure 1 shows the accuracy achieved by three state-of-the-art WFP attacks: CUMUL [8], k-FP [4], and DF [9]. Our experiments confirm our claim that WFP attacks become more effective when the number of consecutively observed pages belonging to a single website increases. We further notice that only two clicks within a site are sufficient to achieve 100% accuracy by using the best performing classifier DF. In a more realistic scenario, already three clicks within a website are sufficient to boost the accuracy of k-FP and CUMUL by approximately 20%, while DF correctly classifies all websites when the user consecutively visits five pages.

**HMM-based.** We further analyze whether the exact sequence of visited pages can improve the overall accuracy achieved by the classifiers. To this end, we combine the worst-performing classifier CUMUL with our HMM-based strategy. First, we assume that the adversary knows all sessions, i.e., the HMM contains transition information for all possible user sessions. In other words, we apply a 10-fold CV where for each page we use 18 traces for training and two for testing. As shown in Figure 2, the accuracy increases significantly for all sessions of length of two or higher.

Next, we evaluate a scenario, in which the user session used for testing is unknown to the attacker, i.e., we use a 10-fold CV to the number of user sessions. In most of the cases, this scenario leads to testing on pages that are also unknown to both CUMUL and HMM. The accuracy decreases slightly compared to the scenario when the sessions are known (see Figure 2). Still, we see a similar positive trend from the use of sequence of pages. The negative effect of unknown sessions is prominent for short sessions and decreases when using longer sessions. For a session of length ten, the accuracy is 96.8% compared to 97.8% when all sessions are known.

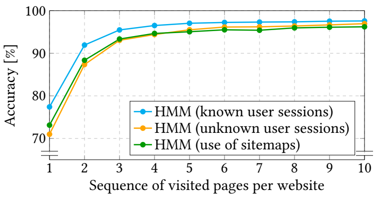

**Figure 2: Accuracy achieved by our HMM-based strategy.**

Finally, instead of learning transition probabilities from user sessions for HMM, we use the sitemap graphs. Although we cannot avoid a potential bias due to the use of the sitemap graphs to collect user sessions, we argue that we can cover orders of magnitude more user sessions by using this approach. As shown in Figure 2, the use of sitemap graphs to compute the transition probabilities is beneficial for user sessions of at most four pages. Still, the difference in the accuracy in the different use cases is neglectable.

To sum up, our results show that our revised, more realistic attacker model for WFP attacks is far more dangerous for users, who consecutively browse multiple pages of a single website.

## 5 CONCLUSION

In this work, we considered a more realistic attacker model for WFP attacks and proposed two strategies that use implicit knowledge on browsing behavior. Our evaluation shows that two, at most three, clicks within a website are sufficient to significantly improve state-of-the-art WFP attacks. We demonstrate that WFP attacks pose a significantly more serious threat to online privacy of Tor users who browse multiple pages of a given website. Our results underline the threat of an attacker who is able to fingerprint a complete website.

As next steps, we will increase the scale and scope of our strategies to identify ways to improve their overall performance. We will also extend our analysis to an open-world setting and study the efficiency of WFP defenses against this new attack surface. While our preliminary analysis shows that the exact sequence of visited pages is only secondary, we would like to further explore its impact.

**Acknowledgments.** Parts of this work have been funded by the EU and state Brandenburg EFRE StaF project INSPIRE and the German Federal Ministry of Education and Research (BMBF) under the project KISS KI Simple & Scalable.

## REFERENCES

[1] Xiang Cai et al. 2012. Touching from a distance: website fingerprinting attacks and defenses. In *ACM CCS*.
[2] Roger Dingledine et al. 2004. Tor: The Second-Generation Onion Router. In *USENIX Security Symposium*.
[3] Martin Ester et al. 1996. A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. In *KDD*.
[4] Jamie Hayes and George Danezis. 2016. k-fingerprinting: A Robust Scalable Website Fingerprinting Technique. In *USENIX Security Symposium*.
[5] Ravi Kumar and Andrew Tomkins. 2010. A Characterization of Online Browsing Behavior. In *19th International Conference on World Wide Web*.
[6] Brad Miller et al. 2014. I Know Why You Went to the Clinic: Risks and Realization of HTTPS Traffic Analysis. *PETS*.
[7] Se Eun Oh et al. 2021. GANDaLF: GAN for Data-LimitedFingerprinting. *PETS*.
[8] Andriy Panchenko et al. 2016. Website Fingerprinting at Internet Scale. In *NDSS*.
[9] Payap Sirinam et al. 2018. Deep Fingerprinting: Undermining Website Fingerprinting Defenses with Deep Learning. In *ACM CCS*.