A Measurement of Genuine Tor Traces for Realistic Website Fingerprinting

Rob Jansen[1], Ryan Wails[2], and Aaron Johnson[1]

[1] U.S. Naval Research Laboratory
{robert.g.jansen7.civ,aaron.m.johnon213.civ}@us.navy.mil
[2] Georgetown University rsw66@georgetown.edu

Abstract Website fingerprinting (WF) enables an adversary to predict the website a user is visiting, despite the use of encryption or Tor. Previous work almost exclusively uses *synthetic* datasets to evaluate the success of WF attacks. We present GTT23, the first dataset of *genuine* Tor traces, intended especially for WF analysis. We obtain it through a measurement of the Tor network, and, with 1.4×10^7 traces, it is larger than any existing WF dataset by an order of magnitude. We survey 28 WF datasets published since 2008 and compare them to GTT23, discovering common deficiencies of synthetic datasets for drawing conclusions about the WF effectiveness. We have made GTT23 available to other researchers.

Keywords: website fingerprinting · traffic analysis · anonymity network

1 Introduction

Website fingerprinting (WF) is a dangerous attack on web privacy because it enables an adversary that can observe a user's outgoing connections to predict the website the user is visiting [6, 18, 19, 26, 47], even if those connections are protected with encryption, virtual private networks (VPNs), or anonymizing networks such as Tor [15]. WF attacks are particularly serious against Tor because they can break Tor's anonymity [5, 9–11, 13, 17, 21, 32–35, 38–41, 43, 44, 52, 53, 55]. In WF on Tor, an adversary guesses the user's destinations from a vantage point that observes the user. The state-of-the-art WF attacks use machine learning (ML), where a classifier is trained using labeled traffic traces to identify the destination. In WF research, labeled data is thus useful for evaluating attack accuracy, both for training and testing.

Through a survey of 28 WF datasets published since 2008 (see §4), we find that all but a *single* prior study consider an adversary that collects labeled traces using an automated browser that programmatically fetches a set of selected webpages through Tor [2]. Such *synthetic* datasets have been criticized as unrepresentative of genuine Tor traffic along numerous axes [21, 25, 36, 40], and their use has led WF research to fall victim to several common ML evaluation pitfalls such as the base rate fallacy [3, 10, 25].

In an effort to address the serious limitations of synthetic WF datasets, a recent study by Cherubin et al. [10] considers a WF strategy in which the

adversary uses a Tor exit relay to collect *genuine* traces, which can be observed and labeled by a relay in the exit position. Genuine traces exhibit the real-world diversity in all factors that might influence classifier performance, and they enable researchers to more accurately evaluate the WF performance that we expect an adversary might realistically attain. Unfortunately, this prior study was done in a *fully online* setting in order to avoid persistently storing genuine data or trained WF classifiers. As a result, it is impossible to replicate their results, and it is difficult to build on the methodology. Indeed, many later works have continued to study WF using synthetically generated datasets [4, 13, 21, 27, 29, 41].

In this paper we present GTT23, the first dataset of labeled *genuine Tor traces*. We describe a large-scale Tor relay measurement plan that we designed to prioritize safety and privacy, which we developed through consultation with our organization's Institutional Review Board and with the Tor Research Safety Board [50] (details on the safety measures appear in App. A of the full version of this paper [22]). We executed our reviewed measurement process to safely measure 13,900,621 circuits to 1,142,115 unique destination domains and 68 unique destination server ports during a 13-week measurement period. We analyze GTT23 and find that 96% of the measured circuits use ports 80, 8080, or 443 to first connect to a destination, that most of the measured circuits carry fewer than 25 cells (<10.5 KB), and that just a single circuit was measured for over 80% of the measured domains. Our analysis of GTT23 helps demonstrate the high degree of traffic diversity with which a WF adversary must contend when launching WF attacks in the real world.

We further evaluate GTT23 to compare its genuine characteristics to those of existing synthetic WF datasets. First, we survey 28 WF datasets published since 2008 and identify several common deficiencies of synthetic datasets. We find that synthetic datasets are composed of a single traffic type (web) using simplistic user models and static software tools while focusing on website *index pages* at uninformed base rates. Second, we conduct a detailed analysis of the statistical disparities between GTT23 and two recent synthetic datasets that are specifically designed for more complex *website* fingerprinting wherein a website contains multiple accessible webpages. We find that the circuit-length variation and website base rates are still not reflected well in the synthetic datasets despite the improved modeling.

We conclude that, because GTT23 contains genuine traces of websites accessed by real Tor users at natural base rates, it is more realistic than any existing synthetic dataset, and thus enables WF evaluations that more accurately estimate real-world WF performance. We also note that, while GTT23 was designed to facilitate WF research, it may be useful for other research on Tor traffic analysis, such as correlation attacks [31] or malware detection [16].

This dataset has been available to researchers upon request since 2024 [23]. This report contains details and analyses of the data to further the understanding and use of GTT23 and to promote the development of similar datasets. A full version of this paper is available with additional details [22].

2 Methodology

2.1 Background

Tor [15] uses onion routing to anonymize TCP connections on the Internet. The Tor network consists of a globally distributed set of *relays*. Each connection through Tor to an outside server is sent through a three-hop *circuit*. This design is intended to prevent an adversary observing any single relay, or one observing either the client or destination but not both, from being able to identify both the source and destination of a connection. In the Tor network today, there are currently over 8,500 relays and over 3 million daily users [49].

To use Tor, a client builds circuits, each passing through an *entry*, a *middle*, and an *exit* relay. A circuit supports multiplexing multiple *streams* of end-to-end TCP communication with internet services. When a new TCP connection to a service is requested by an application (e.g., Tor Browser), the Tor client will use fixed-size application-layer control messages called *cells* to instruct the exit relay to (1) resolve the service's domain name, and (2) make a TCP connection to the service. Each of a circuit's TCP byte-streams is subsequently forwarded bidirectionally through the circuit in data cells. Circuit traffic observed from a single network location can be represented as a time-ordered sequence of (direction, time) pairs (one for each cell sent through a circuit), called a *cell trace*.

A TCP stream may be assigned to any circuit with an exit relay that allows connection to the destination's IP address and port; if no such circuit exists, a new one is built after choosing an exit independently at random from among those with conforming exit policies and weighted by relay bandwidth to balance load. However, Tor Browser employs additional stream assignment rules. When loading a webpage URL, Tor Browser computes the URL's first-party domain name (FPDN) and instructs the Tor client to assign all streams created to load that URL (including those to third-party domains to load embedded objects) on a circuit uniquely associated with the FPDN and isolated from other streams.

Browsing to a page of a new website in Tor Browser will result in a unique FPDN and a new circuit that first resolves a DNS query for the FPDN and then loads the page, while subsequent subpages of that website will be loaded through the same circuit. Cherubin et al. [10] recognized that (1) the FPDN in the circuit's first DNS query can be used to *label* the website of a circuit's cell trace, and (2) an adversary running exit relays can observe *genuine* cell traces and their domain name labels, which can be used to train WF classifiers and produce more realistic estimates of WF performance. (Non-exit relays can observe cell traces but not domain names due to onion routing [48].) Unfortunately, their study considered an online setting to avoid persistently storing sensitive data or classifiers; thus a new measurement is needed to build on the methodology.

In website fingerprinting, an adversary uses the volume and timing of traffic to infer which website is being visited [25]. This attack is suited to breaking the privacy of traffic sent through VPNs or Tor because their traffic plaintext and destination are unobservable. In a WF attack, the adversary observes the client and its traffic. Typical WF attacks use machine-learning classifiers trained on

traffic traces labeled with the destination website (e.g., [11, 41, 43]). The classifiers are applied to traffic traces from the target client to identify the destination website. In the Tor setting, unlike for VPN traffic, WF classifiers are typically only given when a cell appears and in which direction because of Tor's fixed-size cells, which can either be recorded directly by a malicious Tor relay or reconstructed from TCP packet payloads by a network observer.

2.2 Measurement Process

We designed a measurement process that employs one or more Tor exit relays to safely measure genuine Tor cell traces and FPDN labels. The traces and labels are collected into a *dataset* for subsequent analysis. Each participating relay runs a patched version of Tor that we modified to support our measurement as follows.

Circuit Selection When a relay observes a new circuit, it rejects any non-exit type circuit (i.e., onion-service and internal circuits) from measurement. Additionally, the relay applies a probabilistic sampling procedure such that 80% of exit-type circuits are rejected during *high-volume* measurement intervals, and 98% of exit-type circuits are rejected during *low-volume* measurement intervals. Sampling helps us limit the total amount of data collected and provides plausible deniability: any individual circuit created through a participating relay is unlikely to exist in the dataset. Non-rejected circuits are selected for further measurement.

Circuit Measurement A relay internally stores circuit metadata and cell traces during operation for the randomly selected exit-type circuits. To protect some of this metadata, we use the encoding function

$$H(x) = \text{base64encode}(\text{sha256}(x||salt)) \tag{1}$$

where *salt* is chosen uniformly at random, fixed on all measurement relays for the duration of the measurement period, and then destroyed. The relay iteratively constructs a circuit metadata record for each selected circuit (applying $H(\cdot)$ to domain names) until either the circuit closes or N cells have been observed, whichever occurs first.[3] The metadata record is then exported via Tor's control interface to an external process that compresses it, encrypts it with a public-key encryption scheme,[4] and writes it to persistent storage.

Each metadata record includes the following: (1) *day*: an integer number of days that have elapsed since the start of the measurement; (2) *domain*: $H(d)$ where d is the domain name of the circuit's first exit stream;[5] (3) *shortest_private_suffix*: $H(s)$ where s is the shortest private suffix of the pre-image of *domain* computed using Mozilla's public suffix list and `libpsl` [7]; (4) *port*: the server port used when connecting the circuit's first exit stream to its destination; and (5) *cells*: a list of at most N cell metadata items. Each cell metadata item is a 4-tuple containing the time the cell was observed relative to the circuit's creation time,

[3] We use $N = 5{,}000$ cells to remain consistent with previous work.

[4] We encrypt to an offline secret key to prevent on-device decryption.

[5] Circuits for which the first exit stream connects to the destination with an IP address instead of a domain name are rejected from measurement.

an integer encoding the cell's direction, and two integers encoding the cell and relay command, respectively [14].

3 Measurement and Analysis

3.1 Measurement Details

We execute a large-scale Tor measurement study following our methodology from § 2. First, we run a total of eight exit relays, four on each of two identical machines hosted by the Calyx Institute (a nonprofit research and education organization located in NY, USA). Each machine is equipped with 2 Intel Xeon E5-2695 v2 12-core CPUs (48 hyper-threads in total) and connected to an unmetered 1 Gbit/s symmetric network access link. Second, we run a measurement over a 13 week period in 2023; we assign weeks 1, 7, and 13 as high-volume intervals, and the remaining 10 weeks as low-volume intervals. We combined all recorded circuit metadata records into a single dataset which we call GTT23[6] [23].

3.2 Data Analysis

In total, GTT23 contains 13,900,621 circuits, 10,557,898 of which were observed during the high-volume weeks (1, 7, and 13) and 3,342,723 of which were observed during the remaining 10 low-volume weeks.

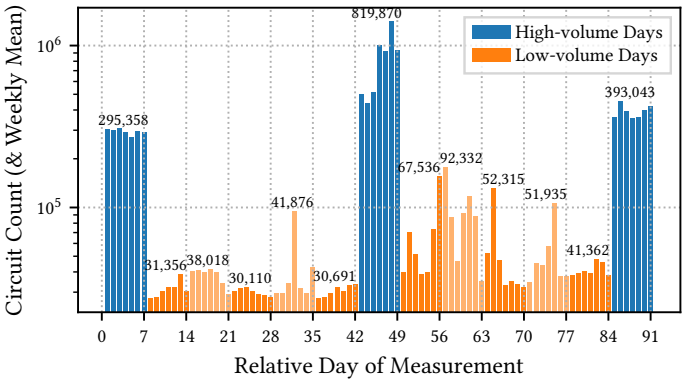

Figure 1: The daily total (bars) and weekly mean (text) number of circuits during our 13 week measurement.

The daily total and weekly mean number of GTT23 circuits are shown in Fig. 1; the daily mean during high-volume weeks is 502,757 and the daily mean during low-volume weeks is 47,753. We observe a slight increase in circuit counts during the latter half of the measurement period which we attribute to natural fluctuation in network usage and the load-balancing weights used for relay selection.

[6] GTT: an acronym for "Genuine Tor Traces"; 2023: the year of measurement.

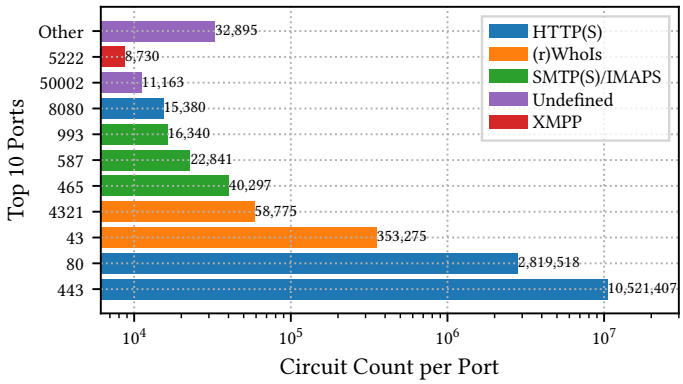

Figure 2: The total number of GTT23 circuits by server port, with IANA-assigned service names [51].

GTT23 contains circuits measured across 68 unique destination server ports. The distribution of the number of measured circuits across the top-ten most-popular service ports is shown in Fig. 2 (with a logarithmic x-axis, and the IANA-assigned service names shown in the legend). We observe that 13,356,305 circuits (96%) use ports 80, 8080, or 443 to connect their first stream to a destination service; these ports are assigned to HTTP and HTTPS by the IANA [51]. The vast majority of the remaining circuits use port 43 or 4321, which are respectively assigned to WhoIs and Remote WhoIs services by the IANA. Frequent connections to these ports have been observed in prior studies of Tor exit traffic [45, 46]: Sonntag observed that they corresponded to a large number of reverse DNS lookups scanning several large networks [45].

The cumulative distribution of the number of observed cells per GTT23 circuit is shown in Fig. 3. We were surprised to find that most circuits are extremely short: the median number of cells over all circuits is just 25, which would support at most 10.5 KB of application payload after accounting for control cells and cell-header overhead. For comparison, we also plot in Fig. 3 the circuit length distribution for the subsets of circuits containing at least 25, 100, and 1,000 cells, respectively corresponding to 10.5, 47.8, and 496 KB of application payload. For reference, the HTTP Archive reports that over 90% of webpages have a transfer size greater than 450 KB across samples of 12 and 16 million desktop and mobile URLs, respectively. Thus, we believe that most GTT23 circuits did not carry full webpage transfers.

GTT23 contains circuits measured across 1,142,115 unique destination domains. The distribution of the number of measured circuits per domain is plotted in Fig. 4. We observe a close fit to a power-law distribution (shape=0.023, loc=0.769, scale=1,495,234), where few popular domains dominate the measurement while a long tail exists with just a single circuit measured for 908,422 (80%) of the domains.

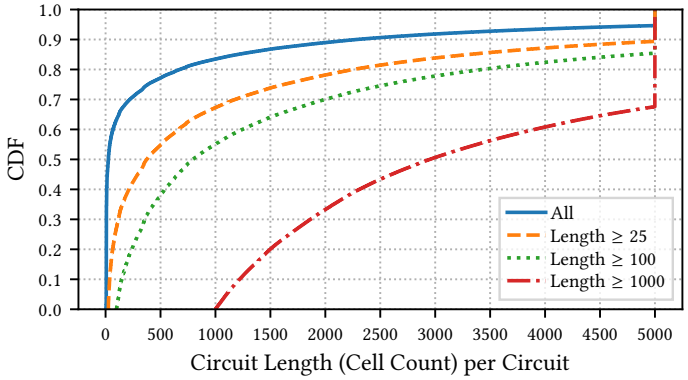

Figure 3: Cumulative distribution of the number of cells per circuit over subsets of GTT23 circuits.

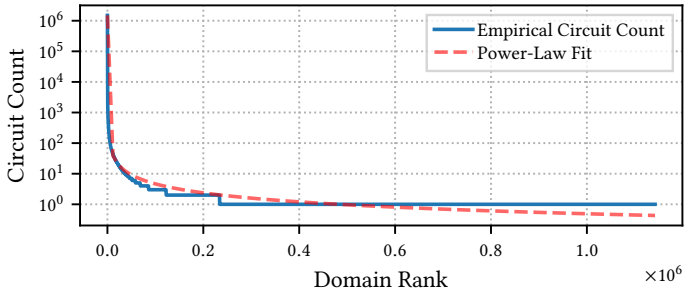

Figure 4: The number of GTT23 circuits per domain; we observe a close fit to a power-law distribution.

Note that obtaining realistic base rates for the domains visited by Tor users is a major advantage of GTT23 over synthetic datasets. In open-world binary classification, the negative class is composed of traces to all sites other than the monitored ones. Thus, the false-positive rate, which is crucial for estimating precision [52], depends on the base rates in the negative class. Similarly, in a multiclass setting (open or closed world), overall WF accuracy depends on the base rates of each class.

Fig. 5 shows the cumulative distribution of two measures of circuit length variability for each domain with more than one GTT23 circuit. The median Coefficient of Variation (i.e., the standard deviation divided by the mean) shows that more than half of the domains have a circuit length standard deviation greater than the mean, while the Coefficient of Dispersion (i.e., the variance divided by the mean) shows that most domains have a relative variance in circuit lengths of multiple hundreds of cells. The high variability in circuit lengths is

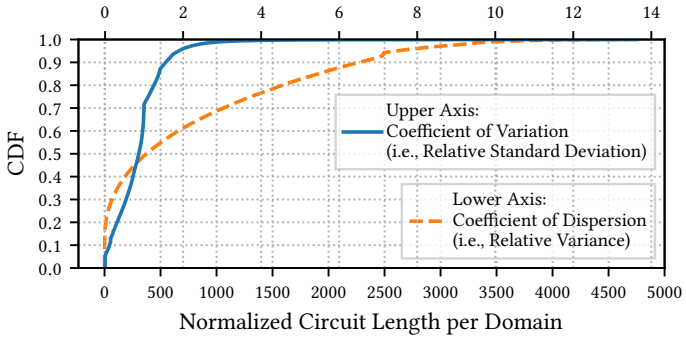

Figure 5: Cumulative distribution of circuit length variation across domains with at least two GTT23 circuits.

consistent with our prior observation that most of the measurement circuits are short, and suggests that many Tor circuits may completely or prematurely fail.

4 Evaluation

In this section, we compare GTT23 and synthetic datasets to understand how well the synthetic datasets model some of the genuine data characteristics that are important for WF.

4.1 Deficiencies of Synthetic Datasets

We survey 28 datasets proposed for WF tasks covering the years 2008–2025. In Table 1 we provide an overview of the properties of a subset of the surveyed datasets selected for their size, complexity, and frequency with which they are used to evaluate later attacks. See Table 2 in App. A for the full comparison.

Like GTT23, these datasets also consist of Tor traffic traces labeled with a destination domain, and they record traffic that actually transited the Tor network and connected to some third-party server. However, we find that every dataset exhibited similar deficiencies: (1) they consist of only web traffic; (2) they are collected using simplistic user models and static software tools, almost exclusively at the client position; (3) they primarily focus on fetching popular webpages; and (4) they do not contain informed base rates. In contrast, real Tor clients use a wide variety of software and software versions, interact with non-web services, and do more than just non-interactively fetch selected webpages. These deficiencies make it difficult to use existing datasets to draw meaningful conclusions about the effectiveness of a WF attack directed at real Tor users [10, 25].

In comparison, GTT23 is the only dataset with traces sampled from genuine traffic created by real Tor users interacting with real internet services at natural base rates. GTT23 is not limited to only web traffic: it contains traces of different

types of internet activity and supports the evaluation of WF attacks and defenses based on websites' first-party domain names (see § 2.1). These traces better represent the WF problem, where an adversary observes undifferentiated traffic from real users and cannot assume that the traffic is just to index web pages or is even to a website at all. Thus, GTT23 can serve to more accurately evaluate the threat posed by a real-world WF adversary. Moreover, GTT23 is larger than the previous largest dataset (the AWF dataset [40]) by an order of magnitude which is important to assess modern deep learning attacks requiring many training examples. Extended results and analysis from our dataset survey appear in App. A.

Table 1: Select WF Datasets (full details in Table 2)

Dataset	**Year**	**Size**	**Description**[†]
k-NN [53]	2014	1.4×10^4	Web, top index pages
AWF CW_{900} [40]	2017	2.3×10^6	Web, top index pages
AWF Open [40]	2017	8×10^5	Web, top index pages
DF [43]	2018	1.4×10^5	Web, top index pages
GoodEnough [37]	2020	2×10^4	Web, top index pages + subpages
BigEnough [29]	2021	3.8×10^4	Web, top index pages + subpages
Multi-tab [13]	2022	5.7×10^5	Web, top index pages, multiple tabs
GTT23	2023	1.4×10^7	Genuine traffic, real user behavior, visited services, natural base rates

[†] All but GTT23 synthetically fetch webpages using automated tools.

4.2 Genuine and Synthetic Disparities

We analyze the statistical disparities between GTT23 and synthetic datasets to understand dataset quality. We place particular emphasis on the trace features found in prior work [17] to be informative for WF. We focus our analysis on two popular synthetic datasets, BigEnough [29] and GoodEnough [37], that were specifically designed to model web*site* fingerprinting; both datasets contain at least ten pages per website, and so they represent among the highest website diversity of the datasets surveyed.

Dataset Composition The GTT23 dataset contains traces generated from real users interacting with any services accessible via the internet (including non-web services), whereas synthetic datasets such as BigEnough and GoodEnough contain traces generated from automated visits to small number of popular websites. Empirical data from this work and previous work suggests that Tor users do not just visit popular websites. First, Fig. 2 shows that a long tail (≈4%) of GTT23 traces are generated from interactions with hosts not running on known web ports such as 80, 443, or 8080. Second, a privacy-preserving measurement of the Tor network performed in 2018 [28] determined that over 20% of web streams

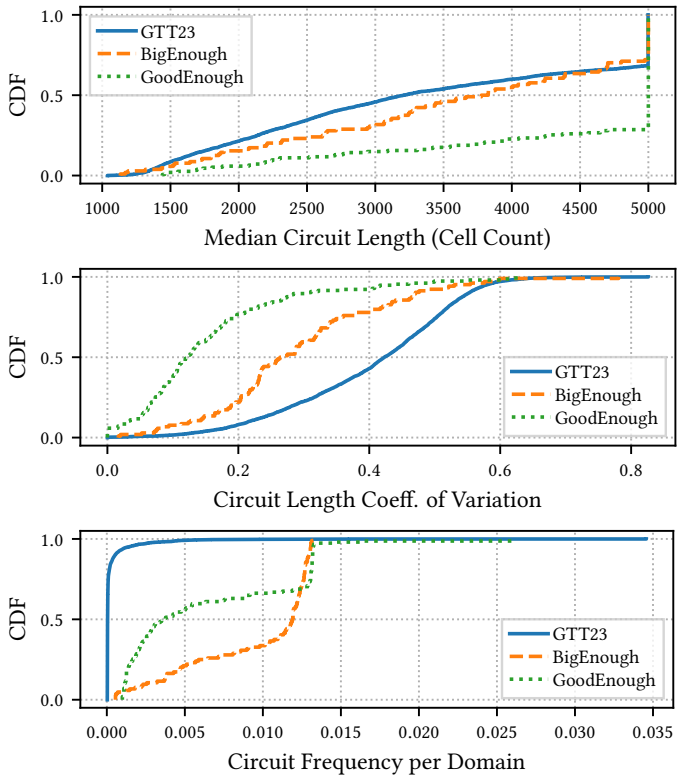

Figure 6: Per-domain statistics computed from the GTT23, BigEnough [29], and GoodEnough [37] datasets. For statistical rigor, here we consider traces with at least 1,000 cells and, among those, domains with at least 30 traces.

exiting the Tor network access a host not in the Alexa Top 1 Million list. This long tail of activity is not reflected in the synthetic datasets and is likely to make the WF classification task more difficult [34].

Data Modeling Even simple features computed from the synthetic datasets do not accurately model genuine Tor traces. Consider, for example, overall trace length, a feature shown in prior work to be informative in the WF task [17]. The top plot in Fig. 6 shows, for all 3 datasets, the distribution of each domain's median circuit length (cell count) for circuit traces with at least 1,000 cells and among those, domains with at least 30 traces. The plot shows that GTT23 traces tend to be shorter than synthetic dataset traces. GoodEnough traces, in particular, tend to be much longer than genuine traces: roughly 70% of GoodEnough domains have a median circuit length of 5,000 cells (the capture limit), whereas this is true of only 32% of domains in the GTT23 dataset. Inaccurate data modeling makes it difficult to draw meaningful conclusions from the synthetic datasets [3].

Intra-class Variance Genuine user traces contain a much richer set of activity than is generated by synthetic, automated crawls to webpages. Genuine traces may be generated from various unpredictable user-initiated behaviors and processes and may reflect complex, interactive sessions with internet hosts, whereas synthetic traces are usually generated by a single, fixed crawling application such as tor-browser-selenium [2] and are limited to simple page accesses. The middle plot in Fig. 6 shows the distribution of the coefficient of variation of trace length—that is, the ratio of the trace length's standard deviation to the mean—for each dataset's domains. At nearly every percentile, the coefficient is higher for GTT23 domains than it is for BigEnough and GoodEnough domains, suggesting that GTT23 traces exhibit higher variation. The higher variance for each domain suggests that WF is more difficult on genuine than synthetic traces.

Base Rates Recall that the frequency of website occurrence in the GTT23 dataset is characterized by a few heavy hitters and a long tail of rarely accessed sites (see Fig. 4). In contrast, the bottom plot of Fig. 6 shows that most domains in the synthetic datasets occur with much higher frequency. For example, the median domain occurs with frequency 5×10^{-4} in GTT23. In comparison, the median domain in the BigEnough and GoodEnough dataset occur with frequencies that are orders-of-magnitude greater, 1×10^{-2} and 4×10^{-3}. Base rate realism is an important aspect of evaluating WF attacks because increasingly low false positive rates are needed to maintain precision at low base rates of occurrence [8, 25, 52]. Precisely fingerprinting most websites in GTT23 requires orders-of-magnitude lower false positives rates compared to BigEnough and GoodEnough.

5 Conclusion

The GTT23 dataset represents the first available collection of genuine Tor cell traces for research in traffic analysis. It has been available to researchers upon request since 2004 since [23], and it has already been used to improve our understanding of WF on Tor. Jansen et al. [24], develop the Retracer methodology for performing WF analysis on Tor trace datasets that, like GTT23 are collected at the exit relay. Retracer modifies the traffic traces to appear more as they would to an adversary observing the client. The results indicate that a WF adversary is likely to obtain much lower accuracy than synthetic datasets have indicated. Jansen [20] further develops this methodology. Similarly, Deng et al. [12] use GTT23 to perform a WF analysis where the adversary is detecting connections to a monitored set of sites. Their results indicate that the Var-CNN classifier obtains higher accuracy than the DF classifier used by Jansen et al.

GTT23 also motivates future work to handle the realities of genuine traces. The appearance of many circuits with few cells requires a WF adversary to consider how much trace data is sufficient to make a confident claim about the destination. The existence of non-trivial amounts of non-Web traffic may motivate new methods to identify the subset of traffic that is to a website at all before applying WF. Accurate base rates may require a WF adversary to choose between training for high accuracy averaged over distinct labels or over traces.

Acknowledgments. This work was supported by the Office of Naval Research (ONR).

References

1. Abe, K., Goto, S.: Fingerprinting attack on Tor anonymity using deep learning. APAN **42** (2016)
2. Acar, G., Juarez, M., individual contributors, tor-browser-selenium - Tor Browser automation with Selenium, `https://github.com/webfp/tor-browser-selenium` (2023).
3. Arp, D., Quiring, E., Pendlebury, F., Warnecke, A., Pierazzi, F., Wressnegger, C., Cavallaro, L., Rieck, K.: Dos and Don'ts of Machine Learning in Computer Security. In: USENIX Security 2022 (2022)
4. Bahramali, A., Bozorgi, A., Houmansadr, A.: Realistic Website Fingerprinting By Augmenting Network Traces. In: ACM CCS 2023 (2023)
5. Bhat, S., Lu, D., Kwon, A., Devadas, S.: Var-CNN: A Data-Efficient Website Fingerprinting Attack Based on Deep Learning. PoPETs **2019**(4) (2019)
6. Bissias, G.D., Liberatore, M., Jensen, D.D., Levine, B.N.: Privacy Vulnerabilities in Encrypted HTTP Streams. In: PET 2005 (2005)
7. C library for the Public Suffix List, `https://github.com/rockdaboot/libpsl` (2023). `https://publicsuffix.org`.
8. Cai, X., Nithyanand, R., Wang, T., Johnson, R., Goldberg, I.: A Systematic Approach to Developing and Evaluating Website Fingerprinting Defenses. In: ACM CCS 2014 (2014)
9. Cai, X., Zhang, X.C., Joshi, B., Johnson, R.: Touching from a distance: website fingerprinting attacks and defenses. In: ACM CCS 2012 (2012)
10. Cherubin, G., Jansen, R., Troncoso, C.: Online Website Fingerprinting: Evaluating Website Fingerprinting Attacks on Tor in the Real World. In: USENIX Security 2022 (2022)
11. Deng, X., Zhao, R., Wang, Y., Zhan, M., Xue, Z., Wang, Y.: Countmamba: A Generalized Website Fingerprinting Attack via Coarse-Grained Representation and Fine-Grained Prediction. In: 2025 IEEE Symposium on Security and Privacy (2025)
12. Deng, X., Chen, J., Yu, L., Zhang, Y., Gu, Z., Qiu, C., Zhao, X., Xu, K., Li, Q.: Beyond a Single Perspective: Towards a Realistic Evaluation of Website Fingerprinting Attacks. Tsinghua Science and Technology (2025)
13. Deng, X., Yin, Q., Liu, Z., Zhao, X., Li, Q., Xu, M., Xu, K., Wu, J.: Robust Multi-tab Website Fingerprinting Attacks in the Wild. In: 2023 IEEE Symposium on Security and Privacy (2023)
14. Dingledine, R., Mathewson, N.: The Tor Protocol Specification, Accessed: September 30, 2023. (2003). `https://gitlab.torproject.org/tpo/core/torspec/-/blob/main/tor-spec.txt`
15. Dingledine, R., Mathewson, N., Syverson, P.F.: Tor: The Second-Generation Onion Router. In: USENIX Security 2004 (2004)
16. Dodia, P., AlSabah, M., Alrawi, O., Wang, T.: Exposing the rat in the tunnel: Using traffic analysis for tor-based malware detection. In: ACM CCS 2022 (2022)
17. Hayes, J., Danezis, G.: k-fingerprinting: A Robust Scalable Website Fingerprinting Technique. In: USENIX Security 2016 (2016)

18. Herrmann, D., Wendolsky, R., Federrath, H.: Website Fingerprinting: Attacking Popular Privacy Enhancing Technologies with the Multinomial Naïve-Bayes Classifier. In: The Workshop on Cloud Computing Security (2009)
19. Hintz, A.: Fingerprinting Websites Using Traffic Analysis. In: PET 2002 (2002)
20. Jansen, R.: CellShift: RTT-Aware Trace Transduction for Real-World Website Fingerprinting. In: NDSS 2026 (2026)
21. Jansen, R., Wails, R.: Data-Explainable Website Fingerprinting with Network Simulation. PoPETs **2023**(4) (2023)
22. Jansen, R., Wails, R., Johnson, A.: A Measurement of Genuine Tor Traces for Realistic Website Fingerprinting, (2024). arXiv: `2404.07892 [cs.CR]`.
23. Jansen, R., Wails, R., Johnson, A.: *GTT23: A 2023 Dataset of Genuine Tor Traces*. Version 1.0.0. 2024. `https://doi.org/10.5281/zenodo.10620520`.
24. Jansen, R., Wails, R., Johnson, A.: Repositioning Real-World Website Fingerprinting on Tor. In: WPES 2023 (2023)
25. Juárez, M., Afroz, S., Acar, G., Díaz, C., Greenstadt, R.: A Critical Evaluation of Website Fingerprinting Attacks. In: ACM CCS 2014 (2014)
26. Liberatore, M., Levine, B.N.: Inferring the source of encrypted HTTP connections. In: ACM CCS 2006 (2006)
27. Limam, Noura, Barradas, Diogo, Arun Naik, Shreya, Singh, Prabhjot, Malekghaini, Navid, A First Look at Generating Website Fingerprinting Attacks via Neural Architecture Search. In: WPES 2023 (2023)
28. Mani, A., Wilson-Brown, T., Jansen, R., Johnson, A., Sherr, M.: Understanding Tor Usage with Privacy-Preserving Measurement. In: ACM IMC 2018 (2018)
29. Mathews, N., Holland, J.K., Oh, S.E., Rahman, M.S., Hopper, N., Wright, M.: SoK: A Critical Evaluation of Efficient Website Fingerprinting Defenses. In: 2023 IEEE Symposium on Security and Privacy (2023)
30. Mitseva, A., Panchenko, A.: Stop, don't click here anymore: boosting website fingerprinting by considering sets of subpages. In: USENIX Security 2024 (2024)
31. Nasr, M., Bahramali, A., Houmansadr, A.: DeepCorr: Strong Flow Correlation Attacks on Tor Using Deep Learning. In: ACM CCS 2018 (2018)
32. Oh, S.E., Mathews, N., Rahman, M.S., Wright, M., Hopper, N.: GANDaLF: GAN for Data-Limited Fingerprinting. PoPETs **2021**(2) (2021)
33. Oh, S.E., Sunkam, S., Hopper, N.: p-FP: Extraction, Classification, and Prediction of Website Fingerprints with Deep Learning. PoPETs **2019**(3) (2019)
34. Panchenko, A., Lanze, F., Pennekamp, J., Engel, T., Zinnen, A., Henze, M., Wehrle, K.: Website Fingerprinting at Internet Scale. In: NDSS 2016 (2016)
35. Panchenko, A., Niessen, L., Zinnen, A., Engel, T.: Website Fingerprinting in Onion Routing Based Anonymization Networks. In: WPES 2011 (2011)
36. Perry, M.: A Critique of Website Traffic Fingerprinting Attacks, (2013). `https://blog.torproject.org/blog/critique-website-traffic-fingerprinting-attacks`
37. Pulls, T.: Towards Effective and Efficient Padding Machines for Tor, (2020). arXiv: `2011.13471 [cs.CR]`.
38. Pulls, T., Dahlberg, R.: Website Fingerprinting with Website Oracles. PoPETs **2020**(1) (2020)
39. Rahman, M.S., Sirinam, P., Mathews, N., Gangadhara, K.G., Wright, M.: Tik-Tok: The Utility of Packet Timing in Website Fingerprinting Attacks. PoPETs **2020**(3) (2020)
40. Rimmer, V., Preuveneers, D., Juárez, M., van Goethem, T., Joosen, W.: Automated Website Fingerprinting through Deep Learning. In: NDSS 2018 (2018)

41. Shen, M., Ji, K., Gao, Z., Li, Q., Zhu, L., Xu, K.: Subverting Website Fingerprinting Defenses with Robust Traffic Representation. In: USENIX Security 2023 (2023)
42. Shen, M., Wu, J., Ai, J., Li, Q., Ren, C., Xu, K., Zhu, L.: Swallow: A Transfer-Robust Website Fingerprinting Attack via Consistent Feature Learning. In: ACM CCS 2025 (2025)
43. Sirinam, P., Imani, M., Juárez, M., Wright, M.: Deep Fingerprinting: Undermining Website Fingerprinting Defenses with Deep Learning. In: ACM CCS 2018 (2018)
44. Sirinam, P., Mathews, N., Rahman, M.S., Wright, M.: Triplet Fingerprinting: More Practical and Portable Website Fingerprinting with N-shot Learning. In: ACM CCS 2019 (2019)
45. Sonntag, M.: Malicious DNS Traffic in Tor: Analysis and Countermeasures. In: ICISSP (2019)
46. Sonntag, M., Mayrhofer, R.: Traffic Statistics of a High-Bandwidth Tor Exit Node. In: ICISSP (2017)
47. Sun, Q., Simon, D.R., Wang, Y.-M., Russell, W., Padmanabhan, V.N., Qiu, L.: Statistical Identification of Encrypted Web Browsing Traffic. In: 2002 IEEE Symposium on Security and Privacy (2002)
48. Syverson, P.F., Goldschlag, D.M., Reed, M.G.: Anonymous Connections and Onion Routing. In: 1997 IEEE Symposium on Security and Privacy (1997)
49. The Tor Metrics Portal, (2023). `https://metrics.torproject.org`
50. The Tor Research Safety Board, (2023). `https://research.torproject.org/safetyboard`
51. Touch, J., Lear, E., Ono, K., Eddy, W., Trammell, B., Iyengar, J., Scharf, M., Tuexen, M., Kohler, E., Nishida, Y.: Service Name and Transport Protocol Port Number Registry, (2023). `https://www.iana.org/assignments/service-names-port-numbers/service-names-port-numbers.txt`
52. Wang, T.: High Precision Open-World Website Fingerprinting. In: 2020 IEEE Symposium on Security and Privacy (2020)
53. Wang, T., Cai, X., Nithyanand, R., Johnson, R., Goldberg, I.: Effective Attacks and Provable Defenses for Website Fingerprinting. In: USENIX Security 2014 (2014)
54. Wang, T., Goldberg, I.: Improved website fingerprinting on Tor. In: WPES 2013 (2013)
55. Wang, T., Goldberg, I.: On Realistically Attacking Tor with Website Fingerprinting. PoPETs **2016**(4) (2016)
56. Zhao, X., Deng, X., Li, Q., Liu, Y., Liu, Z., Sun, K., Xu, K.: Towards fine-grained webpage fingerprinting at scale. In: ACM CCS 2024 (2024)

Appendix

A Survey of Existing WF Datasets

We surveyed prior work related to website fingerprinting attacks in order to better understand the datasets used to quantify attack effectiveness. We evaluated each dataset among a number of different dimensions, as follows.

Year: the time the dataset was collected;
Activity: the kind of user behavior contained in the dataset;
User model: the way in which users perform the activity;

Trace generation software: the tools used in activity creation;
Size: the number of classes and traces in the dataset;
Availability: the accessibility of the dataset to others;
Attacks: the WF attacks originally evaluated on the dataset.

We also noted how each dataset was recorded (that is, the software used and trace observation point, if provided).

The summary of results is shown in Table 2. All datasets surveyed were composed of primarily web activity. Most datasets assume users interact with popular websites, usually those present in the now-discontinued "Alexa Internet" top websites ranking. A few works consider more sophisticated user behaviors: Herrmann et al. [18] collect URLs obtained from monitoring an academic proxy server they had access to; Juárez et al. [25] collect URLs obtained from volunteers browsing the Internet; Panchenko et al. [34] considered URLs obtained from observing Tor HTTP exit traffic, as well as from interacting with popular Internet services such as Twitter and Google; and Deng et al. [13] collected URLs from volunteers browsing the Internet.

The task designated for each dataset may vary. For example, RND-WWW [34], Juárez et al. [25], GDLF-25 [32], GoodEnough [37], BigEnough [29], ALEXA-WSC-FG/BG [30], and CW/OW [56] are designed to incorporate multiple pages for each of many websites. AWF Recollect [40] and WTT-Time [33] are designed to explore aspects of concept drift. DS_{Tor} contains .onion sites in addition to ordinary websites. Multi-tab [13, 56] contain browsing behavior occurring simultaneously in several browser tabs.

All extant datasets are collected synthetically with an automated crawl, often using a single set of software to generate flows (Juárez et al. [25] and Deng et al. [13] both consider the effect that varying versions of Tor Browser Bundle (TBB) may have on attacks). Additionally, nearly every work uses `tcpdump` to collect packet traces on the *client* generation machine. Only GoodEnough, BigEnough, and $D(\text{tbs}, \text{tor})$ [21] collect cell traces using the `tor` process directly; GoodEnough and BigEnough are collected at the client position, whereas $D(\text{tbs}, \text{tor})$ is collected at the guard position.

Inconsistent purposes, over-simplified user models, and static collection software make it difficult to draw meaningful conclusions about the effectiveness of a WF attack directed at real Tor users. Real Tor clients use a wide variety of software (most network applications supporting SOCKS5 can be used with Tor), interact with non-web services, and do more than just non-interactively fetch random pages on the web. In contrast, *GTT23 is the only dataset addressing these weaknesses*—it contains traces from real Tor client interacting with real internet services. Moreover, GTT23 is larger than the previous largest dataset by an order of magnitude (AWF CW_{900} [40]) and is larger than most other existing datasets by multiples orders of magnitude; this volume of data is important when training modern deep learning models which may require millions of examples to be effective.

Table 2: Summary of website fingerprinting datasets curated over the past 15 years. The '⊥' symbol is used to indicate a dataset is unnamed, and the '-' symbol is used when a cell's contents are identical to the above cell. When the year of data collection is not mentioned, we assume it is around ("ca.") the associated article's publication date. Not all datasets describe their trace generation software with the same specificity. N, N_C, N_I, N_{Bg} are the total number of traces in the dataset, the number of positive classes, the number of instances per positive class, and the number of background traces. The "Attacks" column shows a list of WF attack papers evaluated on the dataset.

Ref.	Name	Year	Activity	Activity Detailed	User Model	Trace Gen. Software	N	N_C	N_I	N_{Bg}	Available	Attacks
[18]	⊥ (Hermann)	2008	Web	Links from real-world academic proxy server	Index page	Autofox	8.5×10^3	775	≈10		Dead link	[18]
[9]	⊥ (Cai)	Ca. 2012	Web	Alexa top sites	Index page	tor 0.2.1/2	3.2×10^4	800	≈40		No	[9]
[54]	levdata2	Ca. 2013	Web	Alexa top sites	Index page	tor 0.2.4.7; TBB 2.4.7	4×10^3	100	40		Online	[34, 54]
-	levdata3	-	-	Popular blocked sites, Alexa top sites	-	-	9×10^2	4	10	8.6×10^2	-	-
[53]	k-NN	Ca. 2014	Web	Sensitive sites, Alexa top sites	Index page	TBB 3.5.1; iMacros 8.6.0	1.4×10^4	100	90	5×10^3	Online	[1, 33, 34, 44, 53–55]
[25]	⊥ (Juárez)	Ca. 2014	Web	Alexa top sites, volunteer browsing	Index page, visited pages	TBB (2/3.X); Selenium	4.3×10^4	200	≈40	3.5×10^4	On request	[25]
[55]	⊥ (Wang)	2014	Web	Sensitive sites, Alexa top sites	Index page	tor 0.3.6.4; TBB 3.6.4	9×10^3	100	40	5×10^3	No	[55]
[34]	RND-WWW	Ca. 2016	Web	Twitter, Alexa one-click, Google Trends, Google Random, censored sites	Random subpage	TBB 3.6.1; Chickenfoot; iMacros; Scriptish	1.6×10^5	1,125	40	1.2×10^5	Dead link	[34]
-	TOR-Exit	-	-	HTTP requests of real Tor users	Visited page	-	2.1×10^5			2.1×10^5	-	-
-	WEBSITES	-	-	Popular websites	Index page, random subpage	-	5.3×10^3	50	105		-	-
[17]	DS_{Tor}	Ca. 2016	Web	Alexa top sites, popular .onion sites	Index page	TBB; Selenium	1.1×10^5	85	≈90	1×10^5	Dead link	[17, 33]
[40]	AWF CW_{900}	2017	Web	Alexa top sites	Index page	tor 0.2.8.11; TBB 6.5; Selenium	2.3×10^6	900	2,500		Online	[5, 32, 33, 40, 44]
-	AWF Recollect	-	-	-	-	-	1×10^5	200	500		-	-
-	AWF Open	-	-	-	-	-	8×10^5	200	2,000	4×10^5	-	-
[43]	DF	Ca. 2018	Web	Alexa top sites	Index page	tor-browser-selenium	1.4×10^5	95	1,000	4.1×10^4	Online	[32, 39, 43, 44]
[33]	WTT-time	2018	Web	Alexa top sites	Index page	tor 0.4.0.8; tor-browser-crawler	8×10^4	100	300	5×10^4	On request	[33]
[37]	Good Enough	2020	Web	Alexa top pages, random subpage	Index page	TBB 9.0.2	2×10^4	500	20	1×10^4	Online	
[52]	⊥ (Wang)	2019	Web	Alexa top sites	Index page	tor 0.4.0.1; TBB 8.5a7	1×10^5	100	200	8×10^4	Partially Online	[52]
-	Wikipedia	-	-	Wikipedia browsing	Random subpage	-	2×10^4	100	100	1×10^4	-	-
[32]	GDLF-25	Ca. 2021	Web	Alexa top sites	Random subpage	tor-browser-crawler	9.4×10^4	2,400	39		On request	[32]
-	GDLF-OW	-	-	Links from Rimmer et al. [40]	Random subpage	-	7×10^4			7×10^4	-	-
[29]	BigEnough	2021	Web	Open PageRank top pages	Index page	TBB	3.8×10^4	950	20	1.9×10^4	On request	
[13]	Multi-tab	2022	Web	Alexa top pages	Index page (multi-tab)	TBB; Selenium	5.7×10^5				Online	[13]
[21]	$D(\text{tbs}, \text{tor})$	2022	Web	Wikipedia browsing	Random subpage	tor-browser-selenium	2×10^4	98	200		Online	
[4]	Drift	Ca. 2023	Web	Popular websites, links from Rimmer et al. [40]	Index page	TBB 11.0.10; tor-browser-selenium 0.6.3	1.5×10^4	90	≈110	5×10^3	Online	[4]
	GTT23	2023	Any	Real Tor usage	Visited service	Real client software	1.4×10^7	$\langle\ 1.1\times10^6$	domains	$\rangle$	On request	
[30]	ALEXA-WSC-FG/BG	Ca. 2024	Web	Alexa top sites, random subpage	Random subpage	TBB 7.5.6	8.6×10^5	9,000	90	4.5×10^4	No	[30]
[56]	CW/OW	Ca. 2024	Web	Alexa top sites, random subpage	Random subpage (multi-tab)	TBB	8.1×10^4	1,000	10	9.3×10^3	Online	[56]
[42]	D1–D7	2024	Web	Tranco top sites	Index page	TBB 10.5; Chrome 112.0	7.4×10^5	100	700	4.00×10^3	Online	[42]