Poster: Using Packet Timing in Website Fingerprinting

Mohammad Saidur Rahman, Kantha Girish Gangadhara, Payap Sirinam, Matthew Wright
Center for Cybersecurity, Rochester Institute of Technology
{saidur.rahman, kantha.gangadhara, payap.sirinam}@mail.rit.edu
{matthew.wright}@rit.edu

***Abstract*—Website Fingerprinting (WF) enables an eavesdropper to discover what sites the user is visiting despite the use of a VPN or even the Tor anonymity system. Recent WF attacks on Tor have reached high enough accuracy (up to 98%) to prompt Tor to consider adopting defenses based on packet padding. Defenses such as *Walkie-Talkie* mainly remove features related to bursts of traffic without affecting packet timing. This was reasonable given that previous research on WF attacks ignored or deemphasized the use of packet timing information. In this paper, we examine the extent to which packet timing can be used to facilitate WF attacks. In our experiment, we gained up to 61% accuracy on our unprotected dataset, 54% on our *WTF-PAD* dataset, and 43% on our *Walkie-Talkie* dataset using only timing-based features in an SVM classifier. Using a convolutional neural network (CNN), we got 86% accuracy on our unprotected dataset, 76% and 47% accuracy on our *WTF-PAD* and *Walkie-Talkie* dataset respectively. We intend to investigate further to develop an effective and robust WF attack using packet timing.**

I. Introduction

The Tor anonymity system has more than two million users each day. Tor is, however, vulnerable to traffic analysis attacks. Website fingerprinting is one such attack, and it has received sufficient attention by both researchers [1], [2] and Tor developers [3].

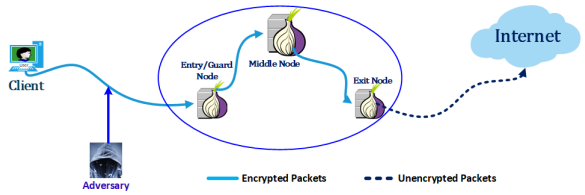

Fig. 1: Network traffic collection

In a website fingerprinting (WF) attack, a passive local eavesdropper (see Figure 1) collects network traffic passing between the client and entry node. From the collected traffic, the attacker then extracts various features and feeds them into a machine learning classifier trained to identify which website the client is visiting. Prior work has shown that this kind of attack is very effective, reaching over 90% accuracy [1], [4], [5].

The Tor Project has given much attention to building defenses against WF attacks [6]. The state-of-the-art attacks emphasize *bursts*, sequences of packets in a single direction (see Figure 2). Because of this, defenses primarily seek to obscure burst patterns. This still leaves the timing of packets as a largely untapped and unprotected resource for features for WF attacks.

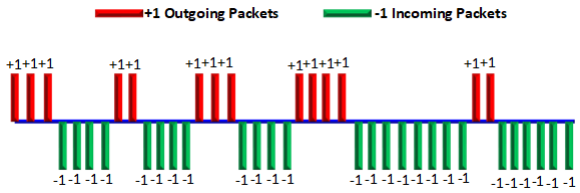

Fig. 2: A visualization of bursts

Prior work on WF attacks discounted timing information [1]. This is because timing characteristics change on each visit to the site, which makes it hard to extract consistent patterns. In this paper, we investigate a novel WF attack using packet timing information. Since individual packet times are generally unreliable as features, we identify more robust timing-related features from the data. Using our new timing features with k-nearest neighbor (k-NN) and support vector machine (SVM) classifiers, we got 51% and 61% attack accuracy, respectively. Using the CNN deep-learning classifier, we got 86% accuracy. These preliminary results indicate that timing information is useful, and it is possible to develop an effective WF attack using timing information. Hence, we intend to investigate more to find the best representations of timing information and develop the attack further.

II. WF Attack using Timing Information

A. Datasets

We use three datasets in our experiments, as shown in Table-I. For undefended Tor traffic and Tor traffic defended with *WTF-PAD* [7], we use the dataset generated by Sirinam et at. [8]. Since the *Walkie-Talkie* defense requires changes to the underlying browser, Sirinam et al. do not have a dataset for that traffic. We thus use the smaller dataset used by Wang et al. [9].

B. Feature Selection and Extraction

We developed several timing-based features that would be more robust from instance to instance of each website than relying on specific packet timings. Since prior work on attacks relies heavily on bursts, we base our timing features on burst-level characteristics. Three of our features are focused on the timing of packets inside a single burst:

TABLE I: Data Sets

Dataset	Source	Classes	Instances in each class	Total
Undefended	Sirinam et al. [8]	95	1000	95,000
WTF-PAD	Sirinam et al. [8]	95	1000	95,000
Walkie-Talkie	Wang et al. [9]	100	100	10,000

TABLE II: *Undefended Tor*: Attack accuracy.

	Features			
Classifier	MED	IMD	IBD	**Combined**
k-NN	41%	20%	18%	51%
SVM	56%	24%	29%	61%
CNN	-	-	-	**86%**

1) Median packet time (*MED*)
2) Variance of the packet times
3) Time from the first packet to the last packet

The other five features consider two consecutive bursts B_1 and B_2:

1) Interval between the medians of B_1 and B_2 (*IMD*)
2) Interval between the end of B_1 and the start of B_2 (*IBD*)
3) Interval between the start of B_1 and the start of B_2
4) Interval between the start of B_1 and the start of B_2, where both B_1 and B_2 are incoming (to the client).
5) Interval between the start of B_1 and the start of B_2, where both B_1 and B_2 are outgoing (from the client).

To create features that would be robust to different instances, we further process the extracted features (see Figure 3). Using the data from all the instances over all websites, we create a histogram of b equal-sized bins for each feature. Then, for each instance, we extract the features from raw data, put them into their respective bins, and take the length of each bin. Finally, we normalize the length of each bin and feed those normalized values into the classifiers.

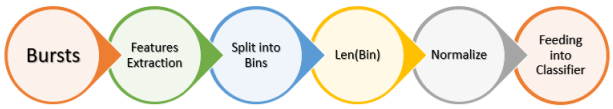

Fig. 3: Processing features

C. Experimental Method and Results

We explored the use of all eight features, both separately and together, in both k-NN, and SVM for different settings of the number of bins b in each histogram. Based on our initial findings, we found that the most effective combination was the three features MED, IMD, and IBD with $b = 20$ bins, so we present results with these features.

As shown in Table-II, MED, IMD, and IBD all provide some classification value on their own, where random guessing would only reach 1% accuracy in this multi-class scenario. Together, these three features can get 61% accuracy with SVM on undefended Tor traffic. Using the CNN classifier, we attain 86% accuracy. While this is not as good as the state-of-the-art classifiers, it is an encouraging result given that we are not using any of the features about bursts cited in prior work as being important for effective classification.

We experimented with the combined feature set against the two defended datasets, using WTF-PAD and W-T, as reported in Table-III. Against WTF-PAD, we attain 54% accuracy using SVM and 76% using CNN, while against W-T, we get 43% and 47%, respectively. Note that 47% is higher than any accuracy results reported on W-T to date and approaching the 50% theoretical maximum accuracy claimed for W-T [9]. That maximum accuracy depends on not using timing data, so we expect that our approach might exceed 50% accuracy when both timing and burst data are considered.

TABLE III: *Defended Tor*: Attack accuracy.

Classifier	WTF-PAD	W-T
SVM	54%	43%
CNN	**76%**	**47%**

III. Conclusion and Future Work

We propose a novel WF attack based on extracting features from packet timing information. We selected and extracted eight new timing features from packet timestamps. We find that our features are robust over multiple noisy instances and provide useful classification value. In the future work, we will investigate more to improve this attack and make it robust. We will also expand our evaluation to Onion services and the open-world setting.

Acknowledgment

This material is based upon work supported by the National Science Foundation under Grants No.1619067.

References

[1] J. Hayes and G. Danezis, "k-Fingerprinting: A robust scalable website fingerprinting technique," in *USENIX Security Symposium*, 2016, pp. 1187–1203.

[2] V. Rimmer, D. Preuveneers, M. Juarez, T. V. Goethem, and W. Joosen, "Automated website fingerprinting through deep learning," in *Network & Distributed System Security Symposium (NDSS)*, 2018.

[3] M. Perry, "A critique of website traffic fingerprinting attacks," *Tor project Blog. Avaliable at: https://blog. torproject. org*, 2013, accessed: 2018-3-28.

[4] A. Panchenko, F. Lanze, J. Pennekamp, T. Engel, A. Zinnen, M. Henze, and K. Wehrle, "Website fingerprinting at Internet scale," in *The Network and Distributed System Security Symposium (NDSS)*, 2016.

[5] T. Wang, X. Cai, R. Nithyanand, R. Johnson, and I. Goldberg, "Effective attacks and provable defenses for website fingerprinting," in *USENIX Security Symposium*, 2014, pp. 143–157.

[6] M. Perry, "Experimental defense for website traffic fingerprinting," *Tor project Blog. https://blog. torproject. org/blog/experimental-defense-website-traffic-fingerprinting*, 2011, accessed: 2018-3-28.

[7] M. Juarez, M. Imani, M. Perry, C. Diaz, and M. Wright, "Toward an efficient website fingerprinting defense," in *European Symposium on Research in Computer Security (ESORICS)*. Springer, 2016, pp. 27–46.

[8] P. Sirinam, M. Imani, M. Juarez, and M. Wright, "Deep fingerprinting: Undermining website fingerprinting defenses with deep learning," *arXiv preprint arXiv:1801.02265*, 2018.

[9] T. Wang and I. Goldberg, "Walkie-Talkie: An efficient defense against passive website fingerprinting attacks," in *USENIX Security Symposium*, 2017, pp. 1375–1390.