UNDERSTANDING FEATURE DISCOVERY IN WEBSITE FINGERPRINTING ATTACKS

Nate Mathews, Payap Sirinam, Matthew Wright

Center for Cybersecurity, Rochester Institute of Technology
{nate.mathews, payap.sirinam}@mail.rit.edu
{matthew.wright}@rit.edu

ABSTRACT

The Tor anonymity system is vulnerable to website fingerprinting attacks that can reveal users Internet browsing behavior. The state-of-the-art website fingerprinting attacks use convolutional neural networks to automatically extract features from packet traces. One such attack undermines an efficient fingerprinting defense previously considered a candidate for implementation in Tor. In this work, we study the use of neural network attribution techniques to visualize activity in the attack's model. These visualizations, essentially heatmaps of the network, can be used to identify regions of particular sensitivity and provide insight into the features that the model has learned. We then examine how these heatmaps may be used to create a new website fingerprinting defense that applies random padding to the website trace with an emphasis towards highly fingerprintable regions. This defense reduces the attacker's accuracy from 98% to below 70% with a packet overhead of approximately 80%.

***Index Terms*—** tor, website fingerprinting, deep learning, convolutional neural network, anonymity

1. INTRODUCTION

Journalists, activists, and privacy conscious individuals look to anonymization software to protect their identify from potential attackers. Among these tools, Tor [1] is perhaps the most popular, with over 8 million users [2]. Tor is a low-latency anonymity network that allows users to conceal their location and browsing behaviors to protect their online privacy. As shown in Figure 1, a client using the Tor Browser passes her traffic through a path of three proxy nodes (the *guard*, *middle*, and *exit*) on the way to her destination website. The connections are all encrypted, which prevents any eavesdropper or compromised proxy node from linking the client with the website or any of the traffic.

Tor is particular popular due to its ability to achieve a high-degree of security while minimizing negative affects to user experience when browsing the internet. The prepackaged Tor Browser Bundle has allowed users with limited technological backgrounds to securely and conveniently browse the world-wide-web, increasing the technologies accessibility to the general public.

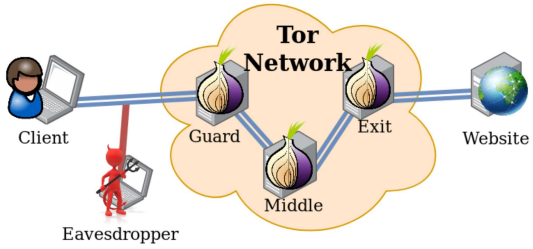

Fig. 1. The Tor website fingerprinting threat model.

Unfortunately, Tor is vulnerable to an attack known as website fingerprinting (WF). In a WF attack, the adversary's goal is to determine what website a Tor user has visited in a browsing session. This attack can be performed by an eavesdropper on the connection between the client and the guard – e.g. an eavesdropper on the the client's wireless link, a compromised cable modem, the user's Internet service provider, or any network along the path. Although all content is encrypted, the attacker can observe patterns in the direction and timing of the network packets produced in the communication between the client and guard. This is often enough to unmask the client's activities and remove the privacy protections provided by Tor. The WF attacker trains a machine learning classifier on the traffic patterns of a number of websites of interest (the *monitored set*) and uses the classifier to predict which site the user is visiting based on the patterns observed on the network. Recently, a particularly potent WF attack that builds on convolutional neural networks (CNN) was introduced. This attack, Deep Fingerprinting (DF) [3], can reach 98% accuracy in a closed-world setting.

To defend against WF attacks, Tor can attempt to change the traffic patterns produced when accessing sites, generally by adding fake packets (*padding*) or delaying real packets. Strong defenses can be prohibitively expensive in terms of the amount of network bandwidth overhead, due to the use of many fake packets, and in terms of latency overhead due to the amount of delay added. Two WF defenses with more realistic overheads have been proposed: WTF-PAD [4] and

978-1-7281-0255-9/18/$31.00

Walkie-Talkie [5]. However, DF is effective against WTF-PAD with over 90% accuracy in closed-world tests. DF also achieved near maximum attacker accuracy against Walkie-Talkie. The effectiveness of this attack highlights the need for a new defense that is robust to WF attacks using deep learning, and is realistic in terms of both overheads and engineering.

In this paper, we explore the use of neural network attribution techniques to better understand why the DF attack is effective so as to develop an opposing defense. These attribution techniques are used to explain what features of the model's input are responsible for the prediction given by the model. When applied to conventional image recognition tasks, these techniques essentially provide heatmaps showing which parts of a trace are most important to the classification decisions made by the network.

Based on our findings with these techniques, we propose a preliminary approach to create a novel WF defense that targets specific regions of sensitivity in the trace. We use neural network attribution techniques to identify the highly fingerprintable regions of a trace. We then apply a new padding scheme to conceal patterns in the traffic, focusing on the highly fingerprintable regions to maximize the impact of fake packets. Our preliminary results show that this defense reduces the accuracy of the DF attack to 66% with acceptable bandwidth and latency overhead.

2. BACKGROUND

WF attacks using handcrafted features. Machine learning-based WF attacks have shown to be effective with over 90% accuracy when classifying webpage traces from closed-world datasets collected by the respective attack's authors. These attacks commonly utilized ML techniques such as k-nearest neighbors (k-NN) [6], support vector machines (CUMUL) (SVM) [7], and random decision forests (k-FP) [8] to achieve these high results. All of these attacks first require an attacker to extract a feature set to be used to train the classifier. Each attack typically requires its own set of handcrafted features be used.

WF defenses. In response to these WF attacks, a series of WF defenses have been proposed. These techniques conceal known features by adding fake packets and/or delaying real packets. Early proposed defenses send packets at a constant rate and insert fake packets such that traffic follows a certain pattern [9, 10]. While these defense have been shown to be effective, they come at the cost of huge bandwidth and latency overheads and are impractical to be deployed in Tor. Recently, two lightweight WF defenses have been proposed and were thought to be good candidates for deployment in Tor due to their efficient overheads. Juarez et al. proposed WTF-PAD [4], which conceals patterns by filling gaps between bursts of traffic with fake bursts without adding delays to real packets. This defense resulted in distortion to the burst features which are commonly used in the engineered feature sets of earlier WF classifiers [4]. Wang et al. proposed Walkie-Talkie [5], a defense that aims to efficiently induce *collisions*—complete overlap in features used by the classifier—between sensitive and non-sensitive sites. WTF-PAD is favored by Tor developers [11], as Walkie-Talkie has significant engineering challenges that have not been addressed. While effective in laboratory experiments, Walkie-Talkie requires significantly changes to the underlying communication protocol as well as advanced knowledge of the traffic patterns of all sites the client might visit [3, 12].

WF attacks using DL. Recently, the breakthroughs of Deep Learning (DL) in many application domains has motivated researchers to apply DL to the WF domain. Abe and Goto [13] were the first to investigate the application of DL for a WF attack. Their attack utilized stacked denoising autoencoders (SDAE) and achieved 88% accuracy in the closed-world scenario. Rimmer et al. [14] studied and evaluated the use of DL for automated features extraction with three different models: SDAE, CNN, and LSTM. They showed that DL can be used to eliminate the feature engineering process with effective results. Their CNN model could achieve a 96% accuracy in the closed-world setting. Most recently, Sirinam et al. [3] proposed Deep Fingerprinting (DF), a sophisticated CNN model that can achieve over 98% accuracy. Their model was the first to undermine the WTF-PAD defense with over 90% accuracy in the closed world.

Neural Network Attribution While DL has shown superior performance on various tasks, it is difficult to explain its classification decisions. This has motivated the study of neural network attribution techniques, which help researchers to understand visually how DL models make decisions. Techniques include Guided-Backpropagation [15], Integrated Gradients [16], DeepLift [17], and GradCAM [18]. These techniques have traditionally been used to identify features and the position of subjects in image classification models. One element of these techniques that makes their application in the WF domain particularly compelling is the ability to generate heatmaps that identify discrimantive regions that highly influence the model's prediction on a particular instance. Knowing these regions would allow a WF defense to focus padding there and lessen the overhead costs of using padding everywhere.

3. WF DEFENSE BASED ON ATTRIBUTION

3.1. Neural Network Attribution in WF

We now examine how different neural network attribution techniques work in the WF setting. We evaluate four techniques: Guided-Backpropagation [15], Integrated Gradients [16], DeepLift [17], and GradCAM [18].

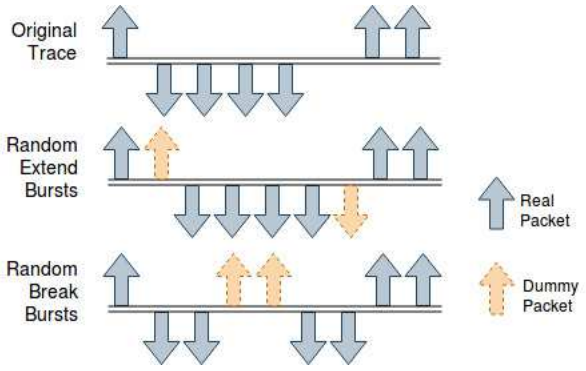

Fig. 2. Visual explanation of Extend Bursts and Break Bursts padding. The direction of the arrow represents the direction of the packet, incoming or outgoing.

When comparing the attribution charts we found that the Guided-Backpropagation, Integrated Gradients, and DeepLift techniques all produced similar, fine-grained explanations. These maps mark the individual bursts and packets that significantly affected the model's final classification. On the other-hand, GradCAM produces coarse-grain maps. These maps instead indicate which regions of the trace contained the features to which the classifier responded positively. As such, maps produced by GradCAM can effectively be used as heatmaps indicating which portion of the trace is more fingerprintable. In our experience, we found that the regional importance captured by GradCAM is naturally applicable to a website fingerprinting defense.

With this understanding, we focused further on the GradCAM charts. To identify the important regions examined by the classifier, we divide all the traces in our dataset into subsets of similar length. For each set, we then average the packet sequence importance scores across trace instances within the same set. This allows us to create one global importance map for each group of similarly sized traces. These maps show which regions in the trace are of global importance to classifying sites.

3.2. Defense Design

The key idea of our defense is simple: we apply random padding to the traffic, where the frequency of the padding is set to match the relative importance of that part of the trace as indicated in the attribution maps. Purely random padding is unlikely to be very effective, however, as that would essentially add a noisy mask over parts of the trace. DL models have been shown to remain effective despite random noise, except when the amount of noise is very high, which would require high levels of padding overhead. Furthermore, some random packets could be detected as obvious padding if they do not follow the usual patterns of web traffic.

Since bursts of traffic have been used in several attacks as the primary basis for creating effective hand-crafted features, we assume that burst characteristics remain important to DL-based attacks. We thus apply padding such that it directly effects bursts in the trace. As in prior work, we define a burst as a set of consecutive packets in a single direction, incoming or outgoing. A website trace can then be thought of a sequence of bursts, outgoing followed by incoming followed by outgoing and so on until the trace ends. To modify a burst B, we can either *extend the burst* by adding padding to B or *break the burst* by adding a fake burst F in the opposite direction that splits B into two real bursts B_1 and B_2, with F in between them.

To evaluate these techniques, we have developed a simulator which operates in two modes: Random Extend Bursts (REB) and Random Break Bursts (RBB). A visual description of these algorithms is shown in Figure 2.

The REB and RBB algorithms evaluate padding on a packet-by-packet basis. In order to determine when padding should be applied, we compare the importance score for the current packet sequence number to a uniformly generated random value between zero and one. If the random value is below the importance score of the packet in the GradCAM chart, a burst of dummy packets is sent. To avoid back-to-back padding, we skip padding on several packets after each evaluation. The number of packets ignored is determined by sampling from a uniform distribution for a selected interval f. When the algorithm decides to pad, the number of dummy packets sent is determined by sampling from a uniform distribution for a selected interval l; the direction of padding is determined by the mode of operation. The optimal intervals for f and l were determined by sweeping a reasonable range of possible values for one while keeping the other constant.

REB can be implemented with minimal additional overhead, since packets need only be added at the end of a burst. For RBB, however, delay must be added during the original burst B to enable the new fake burst F to arrive during the break. Despite being a relatively simple concept, our work is the first to examine this explicit mechanism for a padding defense.

4. EVALUATION

4.1. Dataset

When evaluating the effectiveness of a ML-based attack it is important to discuss the dataset used to train the model. The large dataset collected by Sirinam et al. [3] was chosen for this purpose. This dataset contains traces collected from the homepages of the top 100 websites ranked by Alexa. From each site, 1,250 traces instances were collected. After discarding traces with corrupt traffic, the final dataset includes 95,000 instances from 95 sites.

Like previously proposed defenses, the REB and RBB defenses are executed in a simulator. The simulator was run on

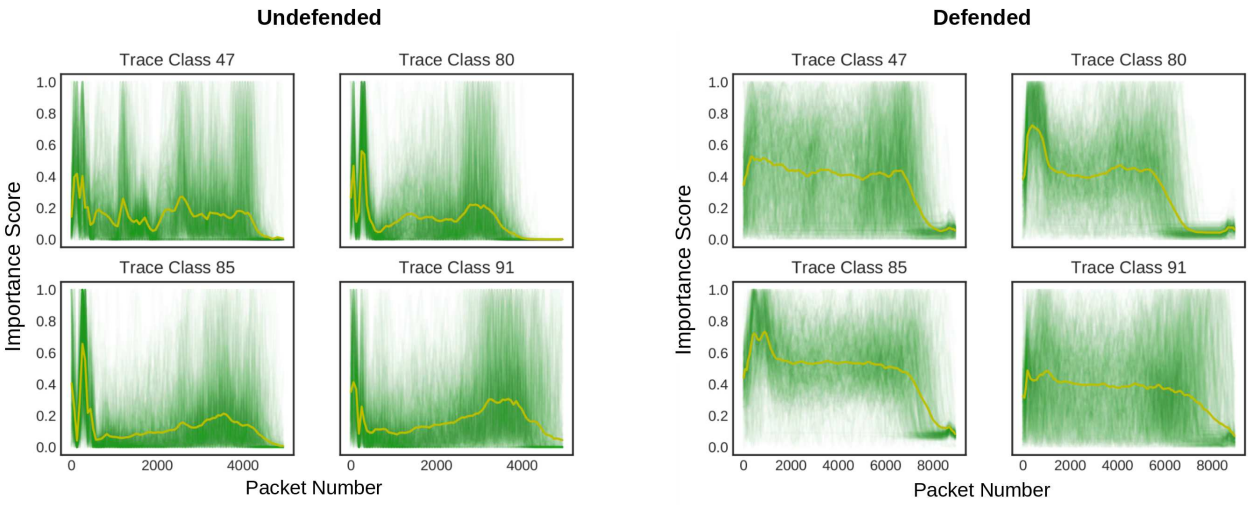

Fig. 3. GradCAM attribution charts for four webpages. Left: importance scores of several undefended trace classes. Right: importance scores of the trace classes with RBB.

Defense	Overheads		Accuracy
	Bandwidth	Latency	
Undefended	0%	0%	98.3%
Tamaraw	328%	242%	11.8%
WTF-PAD	64%	0%	90.7%
Walkie-Talkie	31%	34%	49.7%
REB	83%	0%	95.3%
RBB	83%	28%	66.7%

Table 1. Deep Fingerpinting attack accuracy against WF defense. Results for all expect REB and RBB are sourced from Sirinam et al. [3].

trace instances in the previously described dataset. Given a website trace, our simulator adds fake packets and delays according to the defense's random choices. Delaying a packet by δ leads to adding δ delay to all subsequent packets to ensure that the order remains the same. We note that this may lead to more delay than what would be seen in a real-world implementation.

4.2. Results

We discovered that REB is largely ineffective against the DF model at reasonable overheads. REB can only increase the size of existing bursts; the number of bursts does not change and every large burst remains a large burst in the same sequence as the original trace. On the other hand, RBB changes the burst sequence more dramatically. It not only adds fake bursts to the trace, it can also split long bursts into multiple smaller bursts. Assuming that long bursts are important to the classifier, RBB removes them as a possible feature and masks their location relative to each other. This explains why RBB can reduce classification accuracy significantly with reasonable overheads.

To further understand the effect of our defense on trace patterns, we examine the GradCAM visualizations of several trace classes before and after our defense has been applied. Figure 3 shows the importance scores for four webpages. For each webpage, the importances scores for 800 trace instances are generated and superimposed to show consistency of patterns seen in the trace instances. After the defense has been applied the signal is noticeably more diffuse, indicating that the model has struggled to find consistent patterns in the traces of the same webpage.

Compared to WTF-PAD, RBB is much more effective. However, this security comes at the cost of 19% additional bandwidth overhead and 28% additional latency overhead. We will further explore the trade-offs in security and overheads in future work.

5. DISCUSSION AND FUTURE WORK

In this work, we have presented the design for a novel website fingerprinting defense for Tor that utilizes deep learning attribution techniques to combat the recent success of deep learning based attacks. We find that our Random Break Bursts algorithm is effective against the current state-of-the-art classifier, reducing accuracy from 98% to 66%. These preliminary results suggest that this approach is competitive against other efficient defenses, though further work is still necessary.

At present, we have evaluated our simulated defense under what is known as the closed-world assumption. In the closed-world assumption, the attacker assumes that their tar-

get only visits webpages on which the WF model has been trained. This assumption differs from the more realistic open-world evaluation in which the target may visit any unmonitored website where many webpages are not known by the attacker. Results seen in the closed-world assumption represents the upper-bound of attacker accuracy. Going forward, it will be necessary to evaluate our defense in an open-world scenario. Furthermore, moving this defense out of the realm of simulation to real implementation is necessary to show the full effectiveness of our proposed defense.

Additionally, the defense currently requires knowing the expected trace of the site the client is visiting, much like Walkie-Talkie. We plan to investigate how to perform the defense without this assumption.

6. ACKNOWLEDGEMENT

This material is based upon work supported by the National Science Foundation under Grants No. 1722743 and 1816851.

7. REFERENCES

[1] Roger Dingledine, Nick Mathewson, and Paul Syverson, "Tor: The second-generation onion router," in *USENIX Security Symposium*, 2004.

[2] Akshaya Mani, T Wilson-Brown, Rob Jansen, Aaron Johnson, and Micah Sherr, "Understanding tor usage with privacy-preserving measurement," *arXiv preprint arxiv.org/abs/1809.08481*, 2018.

[3] Payap Sirinam, Mohsen Imani, Marc Juarez, and Matthew Wright, "Deep fingerprinting: Undermining website fingerprinting defenses with deep learning," *ACM Conference on Computer and Communications Security (CCS)*, 2018.

[4] Marc Juárez, Mohsen Imani, Mike Perry, Claudia Díaz, and Matthew Wright, "Toward an efficient website fingerprinting defense," in *European Symposium on Research in Computer Security (ESORICS)*, 2016.

[5] Tao Wang and Ian Goldberg, "Walkie-talkie: An efficient defense againt passive website fingerprinting attacks," in *USENIX Security Symposium*, 2017.

[6] Tao Wang, Xiang Cai, Rishab Nithyanand, Rob Johnson, and Ian Goldberg, "Effective attacks and provable defenses for website fingerprinting," in *USENIX Security Symposium*, 2014.

[7] Panchenko, Fabian Lanze, Andreas Zinnen, Martin Henze, Jan Pennekam, and Klaus Wehrle nd Thomas Engel, "Website fingerprinting at internet scale," in *Network and Distributed System Security Symposium (NDSS)*, 2016.

[8] Jamie Hayes and George Danezis, "k-fingerprinting: A robust scalable website fingerprinting technique," in *USENIX Security Symposium*, 2016.

[9] Xiang Cai, Rishab Nithyanand, and Rob Johnson, "CS-BuFLO: A congestion sensitive website fingerprinting defense," in *Workshop on Privacy in the Electronic Society (WPES)*, 2014.

[10] Xiang Cai, Rishab Nithyanand, Tao Wang, Rob Johnson, and Ian Goldberg, "A systematic approach to developing and evaluating website fingerprinting defenses," in *ACM Conference on Computer and Communications Security (CCS)*, 2014.

[11] Mike Perry, "Padding negotiation," Tor Protocol Specification Proposal. `https://gitweb.torproject.org/torspec.git/tree/proposals/254-padding-negotiation.txt`, 2015.

[12] Sanjit Bhat, David Lu, Albert Kwon, and Srinivas Devadas, "Var-cnn and dynaflow: Improved attacks and defenses for website fingerprinting," *arXiv preprint arxiv.org/abs/1802.10215*, 2018.

[13] K. Abe and S. Goto, "Fingerprinting attack on tor anonymity using deep learning," in *Asia Pacific Advanced Network (APAN)*, 2016.

[14] Vera Rimmer, Davy Preuveneers, Marc Juarez, Tom Van Goethem, and Wouter Joosen, "Automated website fingerprinting through deep learning," in *Network and Distributed System Security Symposium (NDSS)*, 2018.

[15] Jost Tobias Springenberg, Alexey Dosovitskiy, Thomas Brox, and Martin A. Riedmiller, "Striving for simplicity: The all convolutional net," *arXiv preprint arXiv:1412.6806*, 2014.

[16] Mukund Sundararajan, Ankur Taly, and Qiqi Yan, "Gradients of counterfactuals," *arXiv preprint arXiv:1611.02639*, 2016.

[17] Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje, "Learning important features through propagating activation differences," in *International Conference on Machine Learning (ICML)*, 2017.

[18] Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra, "Grad-cam: Visual explanations from deep networks via gradient-based localization," in *International Conference on Computer Vision, (ICCV)*, 2017.