Written Report on "Surakav: Generation Realistic Traces for a Strong Website Fingerprinting Defense", by Gong et al.

Ahmad Charba

April 2023

1 Introduction

In a world where privacy is so easy to compromise, and where looking at what individuals are accessing online tells us more information about them than their closest friends could, it is more than reasonable that people would attempt to regain some of their privacy [1]. Many Privacy Enhancing Technologies have been developed for this purpose, but one of them has stood out since its inception: Tor.

Tor is a circuit based anonymous communication service [2]. It is based on onion routing, a protocol that makes use of nodes to forward encrypted messages to their destination. Tor makes use of onion routing using fixed size packets to allow for near-instantaneous private communication between parties.

Unfortunately, there are still ways that attackers can obtain information about users who use Tor. One of these methods is website fingerprinting (WF), an attack in which a eavesdropper is able to tell which website a user is accessing by using information about packet timings.

This is where trace generators come in: they generate fake traces so that attackers attempting to use website fingerprinting become unable to identify the website the user is actually accessing.

Surakav, proposed by Jiajun Gong et al. in 2022, is one such trace generator. It makes use of a generative adverserial network to mimic normal traffic and minimize the chances of an adversary determining accurately the website a user is accessing without requiring high overhead [3]. The paper in which they proposed Surakav, named "Surakav: Generation Realistic Traces for a Strong Website Fingerprinting Defense", was published in the IEEE Symposium on Security and Privacy.

2 Background

2.1 The Onion Router (Tor) Browser

Tor is a popular privacy enhancing technology that makes use of onion routing to anonymise its users' traffic. The Tor Browser allows users to easily protect themselves, without requiring a lot of technical knowledge [7].

2.1.1 Tor Goals

Tor has been built with the following goals in mind [2]:

- Perfect forward secrecy: being able to decrypt a message does not compromise earlier messages.
- Low latency: Tor is built for low latency communication between parties.
- Deployability, simplicity, flexibility and usability.

2.1.2 Tor Basic Mechanics

Tor relies on onion routing to anonymize its users. It is done in the following way[2]:

1. A set of nodes (let's suppose the common number 3) is selected from the pool of onion nodes.
2. A message from Alice will be encrypted 3 times, in a way that each layer may be decrypted by one and only one node. Each decryption only reveals one recipient; the one which will be able to decrypt the next layer.
3. The message, now encrypted in layers, can be sent to the first node, the entry guard. This node will decrypt the message and obtain the address of the next node.
4. The message is then sent to the following node, the middle node, which will do the same and send the message to the exit node.
5. The exit node will decrypt the message and send it to the target web server.

Now, none of the nodes know both the sender and the receiver of the message. Therefore, the message is dissociated from Alice's identity.

2.2 Website Fingerprinting Attacks

Website fingerprinting is an attack in which a local attacker eavesdrop on network traffic to deanonymize protected web traffic [3]. This type of attack have been getting more and more accurate in recent years, and is now a real threat to Tor users' privacy [4].

2.2.1 Website Fingerprinting Mechanics

An attacker (Eve) can execute a website fingerprinting attack on a Tor user (Bob) in the following manner [5]:

1. Eve visits and interacts with a website (or a list of websites) of interest using Tor. She keeps logs of her own traffic traces: the sent and received packets and their metadata (timestamps, direction etc.).

2. Eve trains a classifier to be able to identify a website based on the packets metadata.

3. Eve then obtain access (or already has access) to Bob's network traffic.

4. Eve keeps logs of Bob's traffic traces for a period of time.

5. Eve uses her trained classifier on Bob's traces to identify if he was accessing one of the websites of interest.

2.2.2 Advancements in Website Fingerprinting

Recent years have show a great increase in the performance of website fingerprinting attacks on Tor [6]. This is a large problem for its users, as they can no longer trust that their browsing activity is unidentifiable by eavesdroppers.

The state-of-the-art technique for website fingerprinting, Tik-tok, which was proposed in 2019 by Rahman et al., is able to achieve very high accuracy, 98.4%, against undefended Tor traffic, 93.5% accuracy on traffic defended using WTF-PAD, and a 64.7% accuracy against onion sites [8].

More recent papers have shown even better results. For instance, in 2022, by using frequency domain fingerprinting, Sun et al. managed to increase the detection accuracy against onion sites by 4.6% when compared to Tik-tok [9]. This shows that website fingerprinting is getting more accurate year over year and that defences against it are a necessity to keep Tor browsing private.

Other notable attacks include:

- k-fingerprinting, which uses random forests to generate fingerprints for website traces,
- CUMUL, which is based on an SVM with cumulative summation of bytes,
- Deep fingerprinting, which uses a deep convolutional neural network that uses sequences of "incoming" and "outgoing" cells as input for the fingerprinting.

2.3 Existing defences against Website Fingerprinting

Many defences have been proposed to counteract Website Fingerprinting. These defences have as a goal to mislead website fingerprinting models, be it by making them unable to distinguish the trace of the browsing activity or by making it look like another website's fingerprinting.

Strategies to counteract website fingerprinting include noise, mimicry, and regularization among others [10]. Many of the proposed approaches involve training a machine learning model, while others utilize traditional methods such as padding and regularizing the packets.

2.3.1 Fake Trace Generation

One of the ways in which a defence will defend against website fingerprinting is by generating fake traces based on the real one. The usual scheme is that a machine learning model would be trained to generate fake packets and timings to send them, making the trace look like it's not coming from the real browsing activity.

One of the goals of these defences are a minimal data and latency overhead, as they will generally be adding and delaying packets to the Tor network. This means that a high data overhead will affect the whole network and its ability to function, while a high latency overhead will make Tor less usable for users. However, this needs to be balanced with the other goal: maximizing the reduction in accuracy for the website fingerprinting attacks.

Some proposals for fake trace generation require information about the website to be known before accessing it, while others use a more general protocol that doesn't require previous knowledge of a website's fingerprint. Other major points of divergence include requiring network cooperation, and the focus of the defence (regularization, randomization and adversarial trace crafting) [**?**].

Some examples of common lightweight defences are WTF-PAD and FRONT. WTF-PAD was broken by Deep Fingerprinting, and FRONT is partially effective against DF and Tik-tok. More heavy defences include Tamaraw.

2.3.2 Other Relevent defences

Walkie-Talkie is a defence that works by merging two traces into a supertrace. It requires two parties to communicate in half-duplex mode, so that they may defend their traces by merging them together, insuring that an attack can only acheive a 50% accuracy at most.

TrafficSilver is a defence that aims to defend against malicious entry nodes, but doesn't grant protection against a local attacker, or one that controls many nodes. It works by routing traffic onto multiple subcircuits, making a single entry node unable to obtain a complete fingerprint.

3 Surakav

Surakav is a trace generator proposed by Jiajun Gong, Wuqi Zhang and Charles Zhang from the Hong Kong University of Science and Technology, and Tao Wang from the Simon Fraser Institute. It is unique in that it makes use of a Generative Adversarial Network (GAN) to generate realistic traces based on some sampled patterns.

3.1 How Surakav Functions

A GAN is a framework in which two neural networks compete. One will play the role of the generator, and the other will play the role of a discriminator. These neural networks will then be trained together: the generator will change to attempt to minimize the accuracy of the discriminator, while the discriminator changes to maximize its own accuracy.

In Surakav's case, the authors have decided to us a Wasserstein GAN (WGAN), a model in which the problem is defined as a min-max problem, loss function is derivated from there. The generator attempts to obtain the minimal accuracy of the best discriminator, which attempts to maximize its own accuracy.

Surakav takes for inputs burst sequences, which are sequences of packets sent or received by the user, and the timings in between the bursts. The generator is first trained with some samples to generate various traces. Then, the generator and discriminator are made to compete with each other: the discriminator goes through training three times for each time the generator is trained. Through the help of an observer, the researcher managed to increase the effectiveness of their method. This observer is a pre-trained model that provides feedback to the generator: the

observer checks whether a fake trace would be classified in the expected class, and this information is fed to the generator.

The trained generator is then used to generate fake traces, which a regulator will use to instruct packet sending for the client and the server. The bursts' size and timings will be adjusted [3].

3.2 Surakav's Results

Surakav was tested on Microsoft Azure as to avoid causing issues in the Tor network. The system was tested against Tik-tok, f-fingerprinting, deep fingerprinting, and CUMUL. Results were compared, in terms of reduction in accuracy and data/latency overhead, against FRONT and Tamaraw, using two presets for Surakav, Surakav-Light and Surakav-Heavy.

The experimental results were that Surakav beats its competing defences in terms of overhead for a similar or better protection. Against Surakav-Light, the best attacks, DF and Tik-Tok, were only able to achieve a true positive rate (TPR) of under 40%, and Surakav-Heavy was able to lower that number to less than 9%. This is while having a much lower data overhead to FRONT and Tamaraw respectively, and a low latency overhead. Front was only able to reduce the TPR to about 43% and deep fingerprinting had a TPR of 15% against Tamaraw.

The authors also looked at the generated traces and compared them to real traces of the aimed class, and found that they were very similar.

3.3 Additionnal Applications

The researchers also showed that Surakav is able to be used to fortify other defences, such as Walkie-Talkie and TrafficSilver. In the case of Walkie-Talkie, they showed that making use of Surakav would make the defence more practical and deployable, by reducing its overhead and increasing the defence's strenght against deep fingerprinting [3].

4 My Critique

Surakav was a very interesting paper to read. I found that the authors gave very sufficient background information about the various concepts necessary to understand Surakav.

The organization of the paper was also well made; information was given before it was needed. The use of an appendix was helpful, as it moves out of the way some

information that isn't required to understand the paper, while keeping it for those who are interested (which helps make the study more reproducible).

The explanations of the way Surakav functions were clear and easy to understand. It gave a good introduction to GANs, and I learned a lot from it.

As for the proposed solution itself, it is innovative in its use of a GAN for generating fake traces, and the proposed method has a very good level of performance for the low amount of overhead it involves. It doesn't have a large number of requirements and seems very deployable.

I found, however, some issues with the paper. For instance, the performance metrics used weren't consistent; in some of their tests, they would use accuracy, while in others, they would specify the TPR and FPR.

It would also have been interesting to see how surakav performs when tuned for a lower latency, with less emphasis on the overhead, as the presented versions add much more latency when compared to other defences such as FRONT.

There is also a general critique of WF defence which can apply here: some authors claim that WF isn't practical, since it loses in accuracy when trying to verify the fingerprint of multiple websites and requires large datasets, among other issues. This is however a debated topic, and the latest attacks seem to show that WF is not slowing down any time soon.

In the end, I would recommend the paper for others to read, as its positive points far outweigh its flaws.

5 Its Place in the Privacy Tree

Surakav, while being an improvement on Tor, which can be put in the leaves 1 and 2; In Tor, Alice attempts to limit exposure by hiding her identity against the intended target and observers.

However, Surakav itself is only aiming to blur the action to external observers. It doesn't help Alice protect her identity. So, Surakav is a PET that prevents exposure to observers (leaf 2) by hiding the action (Alice still wants the server to know what the real packets are). Website fingerprinting generally assumes that the victim's identity is already know (local attacker), so hiding the action in the best course of action to protect against it.

6 A Path Forward

Improvements to Tor, as can be seen by the strength of web fingerprinting attacks, are going to be more and more important going forward. Surakav show the great potential of using GANs to generate fake traces.

Future work could be done on implementing the system in the real Tor network and making it easy to set up and use.

It would also be interesting to see how Surakav compares to other proposed defences that came out in the last year, in term of performance and feasibility, as Surakav was only compared to older technologies.

Testing Surakav against more recent, improved attacks would also be necessary to know if Surakav is worth implementing. WTF-PAD was broken by deep fingerprinting before it was fully implemented, showing the need to test solutions against the latest attacks.

More uses of GANs could also be interesting to see. Similar technologies could be implemented for other types of mix networks, or attempted on image anonymization.

7 Conclusion

In the end, Surakav is a promising technology. It makes good use of GANs to create an effective and practical solution to the privacy risk created by website fingerprinting attacks.

The paper was written in a pleasant manner and opens a path for better defences down the line. It shows the power of GANs for anonymization tasks.

It is my belief that further research on that subject would greatly benefit the security community.

References

[1] Kovila P.L. Coopamootoo. 2020. Usage Patterns of Privacy-Enhancing Technologies. In Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security (CCS '20). Association for Computing Machinery, New York, NY, USA, 1371–1390. https://doi.org/10.1145/3372297.3423347

[2] Dingledine, R., Mathewson, N., Syverson, P.: Tor: The second-generation onion router. In: Proceedings of the 13th USENIX Security Symposium (August 2004)

[3] Gong, J. Zhang, W. Zhang, C. Wang, T.: Surakav: Generating Realistic Traces for a Strong Website Fingerprinting Defense. In: 2022 IEEE Symposium on Security and Privacy (SP), San Francisco, CA, USA, 2022, pp. 1558-1573, doi: 10.1109/SP46214.2022.9833722.

[4] Zhu, Z., Chen, G., Zhang, Z., Fang, M., Song, Q. Mao, B. (2021). Website fingerprinting attack through persistent attack of student. 2021 IEEE 7th International Conference on Cloud Computing and Intelligent Systems (CCIS). https://doi.org/10.1109/ccis53392.2021.9754529

[5] Tao Wang and Ian Goldberg. 2013. Improved website fingerprinting on Tor. In Proceedings of the 12th ACM workshop on Workshop on privacy in the electronic society (WPES '13). Association for Computing Machinery, New York, NY, USA, 201–212. https://doi.org/10.1145/2517840.2517851

[6] W. Juan, C. Shimin, Z. Jun, H. Bin and S. Lei, "Identification of Tor Anonymous Network Traffic Based on Machine Learning," 2021 18th International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP), Chengdu, China, 2021, pp. 150-153, doi: 10.1109/ICCWAMTIP53232.2021.9674056.

[7] S. J. Murdoch and G. Danezis, "Low-cost traffic analysis of Tor," 2005 IEEE Symposium on Security and Privacy (SP'05), Oakland, CA, USA, 2005, pp. 183-195, doi: 10.1109/SP.2005.12.

[8] Rahman et al., M. S., Sirinam, P., Mathews, N., Gangadhara, K. G., Wright, M. (2020). tik-tok: The utility of packet timing in website fingerprinting attacks. Proceedings on Privacy Enhancing Technologies. https://doi.org/10.2478/popets-2020-0043

[9] Sun, Y., Luo, X., Wang, H., Ma, Z. (2022). A method for identifying tor users visiting websites based on frequency domain fingerprinting of network traffic. Security and Communication Networks. https://doi.org/10.1155/2022/3306098

[10] P. Liu, L. He and Z. Li, "A Survey on Deep Learning for Website Fingerprinting Attacks and Defenses," in IEEE Access, vol. 11, pp. 26033-26047, 2023, doi: 10.1109/ACCESS.2023.3253559.