Poster: A Pilot Study on Real-Time Fingerprinting for Tor Onion Services

Young-Ho Kim
Dankook University
Yongin-si, South Korea
dudgh1002@naver.com

Loc Ho
Arkansas State University
Jonesboro, AR, USA
loc.ho@smail.astate.edu

Won-gyum Kim
AiDeep
Seoul, South Korea
wgkim@aideep.ai

Donghoon Kim
Arkansas State University
Jonesboro, AR, USA
dhkim@astate.edu

Doosung Hwang
Dankook University
Yongin-si, South Korea
dshwang@dankook.ac.kr

***Abstract*—Website fingerprinting attacks have exposed a vulnerability in Tor network. Although fingerprinting attacks have shown high success rates, their reality in the real world is still uncertain due to several reasons. To find out the possibility of fingerprinting attacks in real-world environments, we implemented a framework by referring to previous studies. The experimental results show that there is not as much accuracy as expected in classifying many websites, but there is enough accuracy in classifying fewer websites. This pilot study shows the real-time fingerprinting attacks are possible in real world scenarios if a few challenges are addressed.**

I. Introduction

The Tor (The Onion Router) is a Firefox-based anonymous network web service, with more than 1 million users worldwide via secure connection. Tor browser can access both general (non-hidden) websites and onion (hidden) services [1]. The onion services are services that can only be accessed over Tor. Using a Tor browser to access onion services follows a different protocol than accessing general websites. Tor browsers do not receive messages directly from the onion service, but meet at the rendezvous point selected by Tor browser and exchange data [2]. This process requires many steps and involves a lot of data. Such data plays an important role in classifying onion services and general websites. Website fingerprinting attacks using machine learning have exposed a vulnerability in Tor network. The fingerprinting attacks studied earlier show their validity and availability in terms of feature representation, detection rate, capturing large-scale traffic data, and machine learning, but their reality remains uncertain for the real-world practice [3], [4].

This pilot study tries to verify the potentials of Tor website fingerprinting in the real-world. The contributions of this study are as following: (1) We have bulit a framework that can collect network traffic and refine the collected traffic to consist only of Tor-related traffic. (2) We discuss why classifying general websites is more accurate than classifying onion services using the same features. (3) We experiment in a real-time environment provided by the framework.

Network and Distributed Systems Security (NDSS) Symposium 2021
21-24 February 2021
ISBN 1-891562-66-5
https://dx.doi.org/10.14722/ndss.2021.23xxx
www.ndss-symposium.org

II. Related Work

Kwon *et al.* [5] proposed two attack models using the weakness of the hidden service in Tor network. They explained that the number of incoming and outgoing cells and duration of activity can be used as important features to distinguish circuit types. Based on their experiments with 97% accuray, they suggested future possible defenses based on the special properties of the circuits used for hidden service activities. Panchenko *et al.* [3] studied the practical limits of website fingerprinting at Internet scale with more than 300,000 webpages. They used both single webpages (e.g., index.html) and complete websites within realistic internet traffic for the open-world scenario. Their features are the cumulative sum of packets sizes, direction, and ordering. *k*-NN and CUMUL were used for the classifier. To increase the chances of a successful website fingerprinting in the open world, the authors suggested that an attacker would have to crawl many different pages of the site and many instances per page. Although fingerprinting attacks show high detection rates in both closed and open world settings, much research is still needed on whether they can detect in real-time [3], [4].

III. Research Approach

A. Threat Model

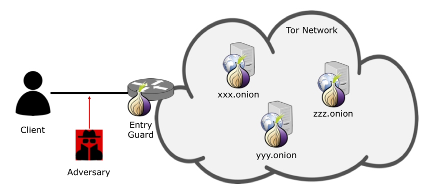


Fig. 1: The Threat Model

An adversary is able to observe the network traffic from a client to the entry Tor router (entry guard) and the traffic from the exit Tor router to a destination client to de-anonymize the connection. Examples of adversaries may be a Tor router owner, ISP (Internet Service Provider), or local network administrator. In this paper, we assume that an adversary monitors the network traffic in the broadcast domain which is between the client and the entry guard (i.e., the first router) as in Figure 1.

B. Data Collection

We developed the network traffic collection system. The system can work in the virtual environment (e.g., XEN Server

Tasks	XGBoost		Decision Tree		Random Forest	
	Accuracy	Training Time	Accuracy	Training Time	Accuracy	Training Time
Binary classification between general and onion	0.9681	2.2474	0.9180	1.032	0.9423	0.4544
All data	0.5682	159.8719	0.4359	2.1927	0.5362	1.2283
50 general websites	0.6841	27.7722	0.5272	0.7746	0.6591	0.4561
38 onion services	0.5135	18.3250	0.3928	0.5466	0.4598	0.3198
8 onion services	0.8914	0.9759	0.8434	0.0452	0.9040	0.0391

TABLE I: Website fingerprinting comparison (Training Time (sec))

and Google Cloud Compute) to minimize network noise and has various features, such as traffic collection scenarios and GUI. The system has several options to collect the most suitable data for the real work environment. For example, the system can determine whether a site should continue to collect traffic and move to the next website, or collecting the sites on the lists once and then repeating them to multiple times. Because the contents of websites can change over time, the performance of the classification may vary depending on how they are collected [6]. Thus, it is important to collect the data that is best suited to the actual environment.

Data were collected from 38 onion services out of 50 candidates that are compiled in `ahmia.fi` [7]. In addition, 50 general websites were collected for comparison with onion services. Each service has 150 instances and the collection time was set to 120 seconds to fully load the websites. We had the list of 50 onion services in February 2020. However, twelve onion (24%) services have been already disappeared in December 2020. The onion services do not provide stable services due to their nature of website contents so some services interrupt or change a URL which is a public key of the onion service.

C. Feature vectors

This work uses two previous works as feature vectors. The first is CUMUL with 104 features which has good results in general website classification as long as the website does not change dynamically [3] . The second is our previous work with 125 features [6]. Originally, the number of features was 103, but 22 were added to this experiment.

IV. Analysis

The classification was conducted with XGBoost, Decision Tree, and Random Forest. Table I shows the results with several classification experiments with 125 features [6]. Binary classification shows high accuracy with 96.81%. The other classifications do not produce good results. We also experimented with CUMUL, but the results were not good. Onion services classification using XGBoost was 42%.

We observed that the classification of onion services is worse than that of general websites despite the same experimental environment (e.g., data collection). We found that the initial common data differs between general websites and onion services. In detail, in onion services' case, the path from the client to the server follows more complex protocols [1]. The onion service requires more data to connect between the client and the server. According to our analysis, the size of the initial common data in the classification of onion services was larger than the size of the initial common data of general websites. We found that such differences led to differences in accuracy in classification.

We still face the challenge of low classification accuracy since Kwon *et al.* [5] showed that the classification accuracy for onion services was more than 98% with 50 onion services. So we analyzed the data to find errors in the process of collecting or analyzing it. Referring to the visualized fingerprints with two popular websites in CUMUL [3], we also used the data we collected to extract CUMUL features and visualized it. Only 8 data were found to follow a similar pattern. The multiple classification was experimented with 8 onion services in a closed-world setting. The accuracy using Random Forest is 90.40% with . Furthermore, experiments were also conducted in open-world scenario. After training the data with less than 1 sec training time, a client accesses onion services in real-time and tried to classify it. The experimental results of the classification accuracy were almost the same as in previous experimental environment (i.e., a closed-world setting). These results indicate that we have the potential to classify onion services in real-time. We still have the difficulty of improving accuracy in many ways, such as eliminating the noise that occurs when data is collected, initial common data, extracting important features, finding the starting and ending points in real-time, etc.

V. Conclusion

This pilot study showed whether real-time fingerprinting attacks are possible in real-time scenarios. The experimental results indicated that real-time fingerprinting attacks are practically possible if a few challenges are resolved. We are now experimenting to classify more websites in real-time, and we will show this result in future work.

Acknowledgment

This material is based upon work supported by the National Science Foundation under Award No. OIA-1946391.

References

[1] Philipp Winter, Anne Edmundson, Laura M Roberts, Agnieszka Dutkowska-Żuk, Marshini Chetty, and Nick Feamster. How do tor users interact with onion services? In *27th {USENIX} Security Symposium ({USENIX} Security 18)*, pages 411–428, 2018.

[2] How do onion services work? https://community.torproject.org/onion-services/overview/. Accessed on January 8, 2021.

[3] Andriy Panchenko, Fabian Lanze, Jan Pennekamp, Thomas Engel, Andreas Zinnen, Martin Henze, and Klaus Wehrle. Website fingerprinting at internet scale. In *NDSS*, 2016.

[4] Rebekah Overdorf, Mark Juarez, Gunes Acar, Rachel Greenstadt, and Claudia Diaz. How unique is your. onion? an analysis of the fingerprintability of tor onion services. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pages 2021–2036, 2017.

[5] Albert Kwon, Mashael AlSabah, David Lazar, Marc Dacier, and Srinivas Devadas. Circuit fingerprinting attacks: Passive deanonymization of tor hidden services. In *24th {USENIX} Security Symposium ({USENIX} Security 15)*, pages 287–302, 2015.

[6] Hyungseok Oh, Donghoon Kim, Won-gyum Kim, and Doosung Hwang. Performance analysis of tor website fingerprinting over time using tree ensemble models. In *2020 International Conference on Computational Science and Computational Intelligence (CSCI 2020)*, 2020.

[7] Tor hidden service search. https://ahmia.fi. Accessed on January 8, 2021.

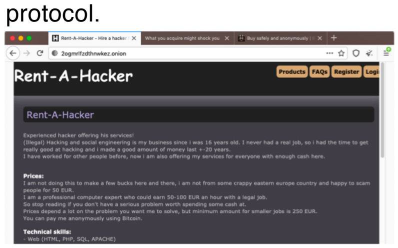
Poster: A Pilot Study on Real-Time Fingerprinting for Tor Onion Services

Young-Ho Kim[1], Loc Ho[3], Won-gyum Kim[2], Donghoon Kim[3], Doosung Hwang[1]
[1]Dankook University, [2]AiDeep, [3]Arkansas State University

Motivation

Website fingerprinting attacks has exposed a vulnerability in Tor Network. Although fingerprinting attacks have shown high success rates, their reality in the real world is still uncertain due to several reasons:

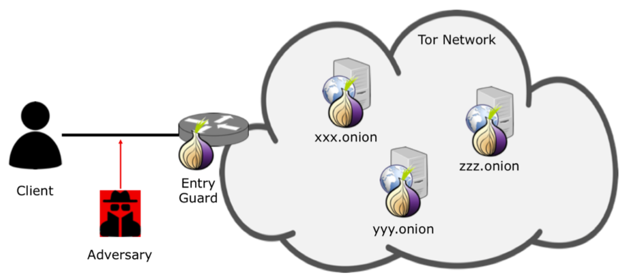
- ❑ There are too many websites in the world.
- ❑ It is not an easy task to find the start point and the end point of a specific traffic in real-time.
- ❑ Accessing onion services and general websites is different due to the protocol.



An example of onion service

Threat Model

An adversary is able to observe the network traffic from a client to the entry Tor router (entry guard) and the traffic from the exit Tor router to a destination client to de-anonymize the connection. Examples of adversaries may be Tor router owners, ISPs, and local network administrators. We assume that an adversary monitors the network traffic in the broadcast domain which is between the client and the first router as in the figure below.



Data & Features

- ❑ 38 onion services and 50 general services
- ❑ Each service has 150 instances.
- ❑ Extracted features

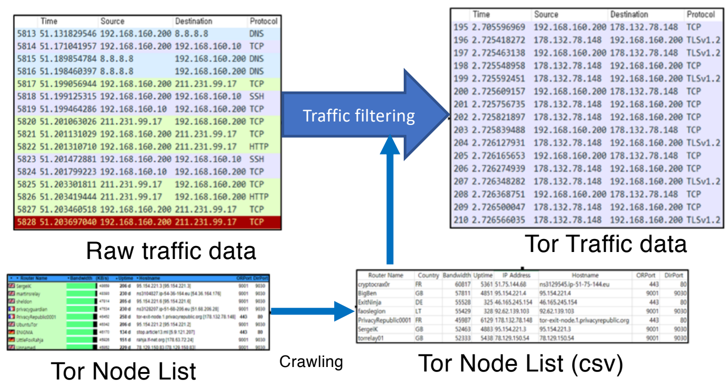
Studied	Feature	No.
CUMUL [8]	Cumulative packet size	104
Oh *et al.* [6]	Packet general information (44) Cell sequence length (4) Packet inter arrival time (27) Burst information (24) Cell ordering (18) Concentration (8)	125

TABLE I: Feature Vectors

Approach

To find out the possibility of fingerprinting attacks in a real-time environment, we implemented a framework.

- ❑ The framework can collect network traffic.
- ❑ The framework can filter the collected traffic to consist only of Tor-related traffic.
- ❑ The frame can monitor the network traffic in real-time.



A part of the framework for filtering Tor traffic

Analysis

The classification was conducted with XGBoost, Decision Tree, and Random Forest with CUMUL and Oh *et al.*'s features

- ❑ The binary classification shows good accuracy with 96.81%.
- ❑ The 8 onion services have 90.40% accuracy.

Classification	XGBoost		Decision Tree		Random Forest	
	Accuracy	TT	Accuracy	TT	Accuracy	TT
Binary classification for all data	0.9681	2.2474	0.9180	1.032	0.9423	0.4544
Label classification for 50 general websites	0.6841	27.7722	0.5272	0.7746	0.6591	0.4561
Label classification for 38 onion services	0.5135	18.3250	0.3928	0.5466	0.4598	0.3198
Label classification for all data	0.5682	159.8719	0.4359	2.1927	0.5362	1.2283
Label classification for 8 onion services	0.8914	0.9759	0.8434	0.0452	0.9040	0.0391

TABLE II: Classification Results (TT: Training Time (sec))

Onion Service URLs
bitstorej4kn3rw3.onion
brohoodahjzxriv7.onion
2ogmrlfzdthnwkez.onion
market7ow7cuw2hz.onion
blackmarthw3vp7a.onion
hupx37mjmbzzw3ja.onion
chemradvzlfaqeqc.onion
poisonj7bdow2nw7.onion

TABLE III: 8 Onion Services

Conclusion

- ❑ This pilot study showed that the real-time fingerprinting attacks are practically possible if a few challenges are resolved.
- ❑ We are now experimenting to classify more websites in real-time, and we will show this result in future work.