BRNO UNIVERSITY OF TECHNOLOGY
FACULTY OF INFORMATION TECHNOLOGY

Excel@FIT 2023
http://excel.fit.vutbr.cz

Fingerprinting Attacks on Anonymity Systems

Bc. Martin Krajči*

Abstract

With the amount of data that is sent daily over the internet, the concept of anonymity has become an important part of everyday life. Despite the fact that most data sent over the network is encrypted, an observer of network traffic can determine who is the packet originator and recipient. In the case that the observed packet is part of a request to a web page, it usually means that the observer can without permission discover what webpages is the given user visiting. For that reason, anonymity systems were created, which aim to disable the linkability of the user's identity and the online activities he is performing. However, attacks, like website fingerprinting attacks, that aim to deanonymize the user often try to prevent this. Many studies tried to verify the effectiveness of this attack, but most of them performed the experiments in an unrealistic environment, which could significantly affect the results. One recent study created list of these mistakes and brought new technologies to improve websites fingerprinting, which will be the subject of this thesis.

*xkrajc21@stud.fit.vutbr.cz, *Faculty of Information Technology, Brno University of Technology*

1. Motivation

While nowadays most of the traffic in the Internet is encrypted, users are still not able to stay anonymous just with the use of encrypted protocols. Packets sent through this network contain personal information, like IP address, which can be used for identification of certain users. That is the main reason why anonymity systems were created. They use different principles to hide users' identities, but in general, they resend users' messages, which means that they appear as originators of that messages.

However, with the appearance of anonymity systems, attacks on these systems also appeared. One of the most powerful attacks on anonymity systems is called fingerprinting attack, which is passive attack and may cause deanonymization of the users [1]. Many previous studies tried to prove severity of this attack by designing and executing different experiments. Accuracy of classification in these experiments was alarmingly high, but their laboratory conditions had unrealistic advantages, which might seriously affect the result.

That is why it is necessary to create experiment under more realistic conditions, which is also goal of this thesis - to experimentaly confirm seriousness of fingerprinting attacks on well known anonymity system - Tor.

2. Fingerprinting attacks on anonymity systems

Fingerprinting attacks can be used in many ways, but after successful execution of this attack, some kind of secret information is obtained, which should not be public. Generally, in the Internet, fingerprinting attacks are used for discovery of what kind of messages are users sending or getting from this network, even though they are encrypted. Visible information, like packet lengths, the time between packets' arrival or server bandwidth can be used for recognition of certain server [2].

But most of this information becomes unavailable, when packets are being relayed through series of nodes, which are in anonymity system Tor called circuit [3]. In such a case, only usable information from the packets is size and their direction. Usability of the size of the packets is also limited, because Tor implemented their own messages, called cells, which are padded to the size of 512 bytes [4, 1]. However, packet size can be used to determine number of cells in each packet, which in combination with direction of each packet creates sequence of cells, in the way

that Tor node seen them. Previous studies introduced method, where cell coming from the client is marked as 1 and cell coming from the server is marked as -1. That creates array of numbers, representing the traffic [5, 6, 7].

However, source codes of Tor node are available, which brings up possibility for another method of data gathering. The source codes could be altered in such a way, that each cell will be directly extracted, which could as well create array of 1 and -1, but it would be more precise then in the previous method.

These gathered arrays will represent each communication with certain server, which in terms of this thesis means requesting and loading of one website on certain web server. These data then can be used to train classificator to determine between sites [5, 6, 7].

3. Attack design

The attack was designed in a similar way to previous studies but with the intention to show impact of realistic configuration and conditions. Mistakes of previous studies include browsing the websites with web browser in default configuration and from the same location, visiting only homepages and gathering training and testing data without realistic time gap.

3.1 Dataset

To show impact of these mistakes, three different datasets were gathered, with the time gap of 5 days in each case, based on list of Alexa top websites. First one was gathered using Chrome web browser in default configuration, which represents data gathered under unrealistic conditions. Second dataset was gathere using also Chrome browser in default configuration, but the client browsing the list of websites was based in different country. Third dataset was gathered using same geolocation as the first one, but the Chrome browser was configured with Spanish locale and Adblock was installed and turned on. These datasets were then used for testing and training N-shot learning classifier, which is relatively new method for classification, requiring much less data then previous methods [8].

3.2 Data gathering scripts

The main script browsing the websites was created using Python, with the help of Selenium [9] and Stem [10] libraries. Thanks to Stem library it is possible to launch Tor process from Python code with special configuration, specifying the Tor guard node that should be used. In this case, our own guard node with altered source codes was specified. The main purpose of Selenium is to communicate with Chrome browser and therefore request it to browse the prepared list of websites. Selenium can also be configured to use created Tor process and relay all the traffic through Tor circuit, containing the Tor node with altered source codes. Besides that, Python script is also used to transfer name of website that is currently being browsed (ground truth) and ID of created Tor circuit to the Tor guard node.

Another Python script was running on the Tor node, which was listening on Linux pipeline for the cells exported from Tor client, and exporting them to files, using the received names of browsed websites.

3.3 Code alteration

To make execution of this attack possible, alteration of source codes in different software was necessary. Source codes of Tor node were altered in such a way, that it could use received circuit ID and export all relay cells which was transferred through this circuit. These cells were exported through the Linux pipeline, where was the Python script listening.

In order to receive global circuit ID, alteration of source codes of Tor control protocol and Stem library was also necessary.

4. Results

While training and testing the classifier with first dataset, which was gathered using Chrome browser in default configuration, the average accuracy was 92%. However, while training the classifier with the first dataset and testing it with second dataset, which was gathered from different geolocation, average accuracy dropped to 38,58%. Similarly, while training the classifier with first dataset and testing it with third dataset, gathered with configured Spanish locale and turned on Adblock, the average accuracy dropped to 39,65%.

5. Conclusion

The designed attack was executed and proved mistakes from previous studies to be very serious, which highly impacted the quality of their results. When comparing results from classification of unrealistic and realistic datasets, it was found out that the accuracy can drop by more then half.

References

[1] Roger Dingledine, Nick Mathewson, Steven J. Murdoch, and Paul F. Syverson. Tor: The

second-generation onion router (2014 draft v1). 2012.

[2] Andrew Hintz. Fingerprinting websites using traffic analysis. In Roger Dingledine and Paul Syverson, editors, *Privacy Enhancing Technologies*, pages 171–178, Berlin, Heidelberg, 2003. Springer Berlin Heidelberg.

[3] Chelsea Komlo, Nick Mathewson, Jacob Appelbaum, Hans-Christoph Steiner, and Taylor Yu. Glossary, 2021.

[4] Roger Dingledine and Nick Mathewson. Tor protocol specification, 2022.

[5] Giovanni Cherubin, Rob Jansen, and Carmela Troncoso. Online website fingerprinting: Evaluating website fingerprinting attacks on tor in the real world. In *31st USENIX Security Symposium (USENIX Security 22)*, pages 753–770, Boston, MA, August 2022. USENIX Association.

[6] Vera Rimmer, Davy Preuveneers, Marc Juarez, Tom Van Goethem, and Wouter Joosen. Automated website fingerprinting through deep learning. 02 2018.

[7] Tao Wang and Ian Goldberg. Improved website fingerprinting on tor. In *Proceedings of the 12th ACM Workshop on Workshop on Privacy in the Electronic Society*, WPES '13, page 201–212, New York, NY, USA, 2013. Association for Computing Machinery.

[8] Payap Sirinam, Nate Mathews, Mohammad Saidur Rahman, and Matthew Wright. Triplet fingerprinting: More practical and portable website fingerprinting with n-shot learning. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, CCS '19, page 1131–1148, New York, NY, USA, 2019. Association for Computing Machinery.

[9] Selenium automates browsers. that's it!, © 2023.

[10] Stem docs.