University College London

Dept. Computer Science

Automatic Feature Selection for Website Fingerprinting

Axel Goetz

BSc. Computer Science

Supervisors:

Dr. George Danezis

Jamie Hayes

March 4, 2017

This report is submitted as part requirement for the BSc Degree in Computer Science at UCL. It is substantially the result of my own work except where explicitly indicated in the text. The report may be freely copied and distributed provided the source is explicitly acknowledged.

Abstract

Contents

1 Introduction

1.1 The Problem

The internet has become an essential tool for communication for a majority of the population. But privacy has always remained a major concern, which is why nowadays most web content providers are slowly moving away from HTTP to HTTPS. For instance, at the time of writing, around 86% of Googles traffic is encrypted. This is a significant improvement compared to 2014 where only 50% was sent over HTTPS [4]. However, this encryption only obscures the content of the web page and does not hide what websites you are visiting or in general who you might be communicating with.

Hence, an Internet Service Provider (ISP) can easily obtain a lot of information about a person. This is an especially large concern for people living in oppressive regimes since it allows a government to easily spy on its people and censor whatever websites they would like. To circumvent these issues, several anonymization techniques have been developed. These systems obscure both the content and metadata, allowing a user to anonymously browse the web. One of the most popular low-latency anonymization networks is called Tor, which relies on a concept called Onion Routing [15].

The list of known attacks against Tor is at the time of writing very limited and most of them rely on very unlikely scenarios such as having access to specific parts of the network *(both entry and exit nodes)* [15]. However, for this report we will make a more reasonable assumption that an attacker is a *local eavesdropper*. By this we mean that the entity only has access to the traffic between the sender and the first anonymization node, like ISPs.

One of the most successful attacks that satisfies these conditions is known as *website fingerprinting* (WFP). It relies on the fact that Tor does not significantly alter the shape of the network traffic [6]. Hence, the attack infers information about the content by analysing the raw traffic. For instance by analysing the packet sizes, the amount of packets and the direction of the traffic, we might be able to deduce which web pages certain users are visiting. Initially, Tor was considered to be secure against this threat but around 2011, some techniques such as the *support vector classifier* (SVC) used by Panchenko et al. started to get recognition rates higher than 50% [11].

However, one of the main problems with majority of the attacks proposed in the research literature, is that they rely on a laborious, time-consuming manual feature engineering process. The reason why is because most primitive machine learning techniques are only able to process fixed-length vectors as its input. Therefore features need to be picked based on intuition and trial and error processes that reveal the supposedly most informative features.

Thus the goal of this paper is to investigate the use of novel deep-learning techniques to automatically extract a fixed-length vector representation from a traffic trace that represents loading a web page. Next, we aim to use these features in existing attacks to see if our model successfully identified the appropriate features.

1.2 Aims and Goals

We can subdivide the project up into different aims, each with their own goals:

1. **Aim:** Critically review the effectiveness of current website fingerprinting attacks.
   Goals:

   - Analyse various models that are currently used in fingerprinting attacks.
   - Examine how would a small percentage of false positives impacts a potential attack.
   - Analyse how the rapid changing nature of some web pages would impact the attack.
   - Review if there are any assumptions that are being made that could impact the effectiveness of an attack.

2. **Aim:** Automatically generate features from a trace that represents loading a webpage.
   Goals:

   - Critically compare various different feature generation techniques such as stacked authoencoders, RNN sequence-to-sequence and bidirectional RNN encoder decoder models.
   - Identify a dataset that is large enough to train a deep-learning model.
   - Compare several software libraries to perform fast numerical computation such as Tensorflow, Torch, Theano and Keras.
   - Implement the most appropriate feature generation model in one of the previously mentioned software libraries.

3. **Aim:** Train existing models with the automatically generated features and test their performance compared to hand-picked ones.
   Goals:

   - Identify five different models that have previously been successful in various website fingerprinting attacks and implement those models.
   - Extract the same hand-picked features from our dataset as mentioned in the respective papers.
   - Investigate an appropriate technique for evaluating the results of different models.
   - Compare the hand-picked features compared to the automatically generated ones for different models. In addition, we also want to investigate their effectiveness in different threat models. For instance if an adversary wants to identify which specific web pages a user is visiting *(multi-class classification)* or if the adversary just wants to know whether the user visits a web page from a monitored set of web pages *(binary classification).*

1.3 Project Overview

As previously mentioned, the general project can be split up into three different aims. Hence, we also approached it in three stages:

- First, we examine different existing website fingerprinting models to gain a deeper understanding of the concept.
- Next, we perform more research to contrast different automatic feature selection models and implement the most appropriate model.
- Finally, we compare the effectiveness of hand-picked features with the automatically generated ones by training them on different existing website fingerprinting models.

1.4 Report Structure

2 Background Information, Related Work and Research

2.1 The Problem

2.1.1 Onion Routing

2.1.2 Machine Learning

2.1.3 Website Fingerprinting

2.1.4 Defenses

2.1.5 Critical Analysis

2.2 Related Work

2.3 Deep Learning

2.4 Automatic Feature Selection

2.4.1 Stacked Authoencoder

2.4.2 Sequence-to-Sequence Model

2.5 Software Libraries

3 Attack Design and Implementation

3.1 Threat Model

3.2 Design Overview

3.3 Sequence-to-Sequence Model

3.4 Website Fingerprinting Models

3.5 Hand-picked Features

4 Evaluation and Testing

4.1 Evaluation Techniques

4.2 Evaluation

4.2.1 Data Collection

4.2.2 Sequence-to-Sequence Model

4.3 Testing

5 Conclusion

5.1 Future Works

Bibliography

[1] Kota Abe and Shigeki Goto. "Fingerprinting Attack on Tor Anonymity using Deep Learning". In: *Proceedings of the APAN Research Workshop 2016* ().

[2] Xiang Cai et al. "Touching from a Distance: Website Fingerprinting Attacks and Defenses". In: *Proceedings of the 2012 ACM conference on Computer and communications security - CCS '12* (2012). DOI: `10.1145/2382196.2382260`. URL: `http://pub.cs.sunysb.edu/~rob/papers/fp.pdf`.

[3] Kyunghyun Cho et al. "Learning Phrase Representations using RNN Encoder Decoder for Statistical Machine Translation". In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (2014). DOI: `10.3115/v1/d14-1179`. URL: `https://arxiv.org/pdf/1406.1078v3.pdf`.

[4] Google. "Google Transparency Report". In: (Mar. 2017).

[5] Xiaodan Gu, Ming Yang, and Junzhou Luo. "A novel Website Fingerprinting attack against multi-tab browsing behavior". In: *2015 IEEE 19th International Conference on Computer Supported Cooperative Work in Design (CSCWD)* (2015). DOI: `10.1109/cscwd.2015.7230964`.

[6] Jamie Hayes and George Danezis. *k-fingerprinting: A Robust Scalable Website Fingerprinting Technique*. URL: `https://www.usenix.org/conference/usenixsecurity16/technical-sessions/presentation/hayes`.

[7] Marc Juarez et al. "A Critical Evaluation of Website Fingerprinting Attacks". In: *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security - CCS '14* (2014). DOI: `10.1145/2660267.2660368`. URL: `http://www1.icsi.berkeley.edu/~sadia/papers/ccs-webfp-final.pdf`.

[8] Christopher Olah. *Understanding LSTM Networks*. URL: `http://colah.github.io/posts/2015-08-Understanding-LSTMs/`.

[9] Barak Oshri and Nishith Khandwala. "There and Back Again: Autoencoders for Textual Reconstruction". In: (). URL: `http://cs224d.stanford.edu/reports/OshriBarak.pdf`.

[10] Andriy Panchenko et al. "Website Fingerprinting at Internet Scale". In: (). URL: `https://www.comsys.rwth-aachen.de/fileadmin/papers/2016/2016-panchenko-ndss-fingerprinting.pdf`.

[11] Andriy Panchenko et al. *Website fingerprinting in onion routing based anonymization networks*. 2011. DOI: `10.1145/2046556.2046570`. URL: `https://www.freehaven.net/anonbib/cache/wpes11-panchenko.pdf`.

[12] Mike Perry. *A Critique of Website Traffic Fingerprinting Attacks*. URL: `https://blog.torproject.org/blog/critique-website-traffic-fingerprinting-attacks`.

[13] Vera Rimmer. "Deep Learning Website Fingerprinting Features". In: (2016). URL: `https://www.esat.kuleuven.be/cosic/publications/thesis-280.pdf`.

[14] Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. "Sequence to Sequence Learning with Neural Networks". In: (). URL: `https://arxiv.org/abs/1409.3215`.

[15] *The Tor Project*. URL: `https://www.torproject.org/docs/faq`.

[16] 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* (2013). DOI: `10.1145/2517840.2517851`. URL: `https://www.freehaven.net/anonbib/cache/wpes13-fingerprinting.pdf`.

[17] Tao Wang et al. “Effective Attacks and Provable Defenses for Website Fingerprinting”. In: *USENIX Security Symposium* 23 (Aug. 2014). URL: https://www.usenix.org/system/files/conference/usenixsecurity14/sec14-paper-wang-tao.pdf.

[18] Zhanyi Wang. “The Applications of Deep Learning on Traffic Identification”. In: (). URL: https://www.blackhat.com/docs/us-15/materials/us-15-Wang-The-Applications-Of-Deep-Learning-On-Traffic-Identification-wp.pdf.