Investigating the Effects of Pre-Fetching on Website Fingerprinting Attack*

Department of Computer Science and Engineering
University of Minnesota

Vaibhav Sharma
sharm361@umn.edu

Taejoon Byun
taejoon@umn.edu

Se Eun Oh
seoh@umn.edu

Elaheh Ghassabani
ghass013@umn.edu

ABSTRACT

This content will be edited later. We plan to explore the area of website fingerprinting in anonymization networks starting with the paper Website Fingerprinting in Onion Routing Based Anonymization Networks.

Keywords

Website fingerprinting, anonymity, encrypted traffic, Tor

1. INTRODUCTION

The content of this section will be changed later. Penchenko et al. [**?**] were among the first to report website fingerprinting attacks with reasonable accuracy on Tor. This paper provides a sufficient understanding of the feature set and classification framework required for this attack. Some of the team members are familiar with data mining techniques and software packages required for their use. All team members have access to the compute servers provided by the Computer Science and Engineering department and will request access to the other high performance computing resources if required.

2. RELATED WORK

This section discusses related work in the area of website fingerpriting. We will update the content shortly [2, 4, 5, 3].

3. BACKGROUND

This section provides a brief description of required background.

*CSCi5271: Introduction to Security

ACM ISBN 978-1-4503-2138-9.
DOI: 10.1145/1235

3.1 Link Pre-fetching

Today's web browsers, including Tor, makes use of a specific syntax called *pre-fetching*, which was proposed as a draft standard by Mozilla. Using pre-fetching, browser can predicts documents likely to be visited by the user in the near future. Therefore, based on the hint provided by pre-fetching a browser is able to fetch those documents a head of time. In fact, it is the web page that provides a set of pre-fetching hints for the browser. Then, loading the page and passing an idle time, the browser starts to pre-fetch and cache specified documents. Needless to say, this mechanism improves efficiency. Particularly, it is most effective if the content provider may be reasonably certain which links users are going to visit next [1].

3.2 Network Analysis and Classifiers

description about how we analyzed the traffic. And which classifer used, which/how features are extracted.

4. EFFECTS OF PRE-FETCHING ON FINGERPRINTING

In this section, we will write about our experiments. We are planning to conduct two sets of experiments. If we consider the network traffic, the number of packages go upstream depends on the number of pre-fetching requests, and the number of downstream packages coming depends on the size of resources that should be pre-fetched. Therefore, it is obvious that pre-fetching would affect the fingerprint of the traffic of a particular website. *should be completed.*

4.1 Investigate Pre-Fetching Effects on top 60 Popular Websites

We are running experiments to see how pre-fetching affects the websites' fingerprints. After doing some search on top popular websites, we put together a small crawler by which we learnt that only around 60 of all 6000 websites are use pre-fetching mechanism. We are capturing traffic of these websites in two different modes: 1) with enabled pre-fetching, and 2) with disabled pre-fetching. We are working on feature extraction, and about to decide which classifiers to use for the learning phase. Ultimately, we plan to conduct two sets of experiments. One sort of experiment is to compare two series of the captured packets and find the accuracy number with the help of a classifier, by which our

goal is to provide an evidence to see if pre-fetching really affects fingerprints of websites. So, if the result will be positive, we will perform another set of experiment, which kind of simulates a sub set of those 60 websites. Then, we will see how (altering) the size of pre-fetching affects fingerprinting attacks/ defense mechanisms.

4.2 Effect of Pre-Fetching Packets Size on Fingerprinting Attacks

Here, we will explain our second experiment. We will simulate a sub set of webpages we investigated in the previous experiment. Then, we will equip them with a mechanism so that they can affect the downstream traffic and finally their fingerprint. Then, we will analyze the result to see how this idea contributes to the effectiveness of attacks and defense techniques.

5. EXPERIMENTS

We are planning to conduct two sets of experiments.

6. CONCLUSIONS

This section will conclude the result of our experiments. Finally we will provide some evidence to show how pre-fetching affect fingerprinting attacks. Based on our result, we are planing to suggest some defense mechanisms.

7. ACKNOWLEDGMENTS

This is a research project for CSCi5271, University of Minnesota.

8. REFERENCES

[1] Link prefetching. https://en.wikipedia.org/wiki/Link_prefetching, 2015. Accessed: 2015-12-04.

[2] K. P. Dyer, S. E. Coull, T. Ristenpart, and T. Shrimpton. Peek-a-boo, i still see you: Why efficient traffic analysis countermeasures fail. In *Proceedings of the 2012 IEEE Symposium on Security and Privacy*, SP '12, pages 332–346, Washington, DC, USA, 2012. IEEE Computer Society.

[3] M. B. G. D. B. Kopf. Preventing side-channel leaks in web traffic: A formal approach. 2013.

[4] A. Panchenko, L. Niessen, A. Zinnen, and T. Engel. Website fingerprinting in onion routing based anonymization networks. In *Proceedings of the 10th annual ACM workshop on Privacy in the electronic society*, pages 103–114. ACM, 2011.

[5] T. Wang and I. Goldberg. Improved website fingerprinting on tor. In *Proceedings of the 12th ACM workshop on Workshop on privacy in the electronic society*, pages 201–212. ACM, 2013.