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

Vaibhav Sharma
sharm361@umn.edu

Taejoon Byun
taejoon@umn.edu

Se Eun Oh
seoh@umn.edu

Elaheh Ghassabar
ghass013@umn.ed

Department of Computer Science and Engineering
University of Minnesota
Minneapolis, MN 55454

ABSTRACT

This paper is written to get an A in the CSCI5271 course. PERIOD.

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. [4] 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

3. BACKGROUND

This section provides a brief description of required background.

3.1 (

Website Fingerprinting

*This report is submitted as a partial fulfillment of *CSCI5271: Introduction to Security* course.

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

3.2 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].

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.

Table 1: Experiment design to answer *RQ1*

victim \attacker	prefetch on	prefetch off
prefetch on	(1)	(2)
prefetch off	(3)	(4)

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

Research Questions

1. **RQ1**: Does prefetching itself provide an extra degree of defense?
2. **RQ2**: Can prefetching be used as a browser-side defense mechanism?

1. We speculate that prefetching itself might provide extra defense because of the extra packets. It can also be the case however, prefetching websites are more vulnerable to fingerprinting because of the extra prefetch packets that shows distinct prefix.
2. This case is unlikely since prefetching is on by default. We assume that victims will more likely be using Tor under the default setting.
3. This case is what we are most curious about, whether a victim can confuse an attacker by simply turning prefetching setting off of his browser.
4. This case simulates a situation where a victim is loading any other websites that does not prefetch any resource. This can be used as a comparison case.

6. CONCLUSIONS

What we have done in this work is so cool and awesome. What you may criticize will all be put here as "future work".

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

7. ACKNOWLEDGMENTS

The authors appreciate Professor Stephen McCamant for telling geeky jokes in classes all the time. 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.