Website Fingerprinting using Traffic Analysis Attacks

Salini Selvaraj Kowsalya

University of Wisconsin, Madison

Abstract. Website fingerprinting is the act of recognizing web traffic through surveillance despite the use of encryption or anonymizing software. The overall idea is to leverage the fact that many web sites have specific request patterns, response byte counts, and other similar coarse features that are known beforehand. This information can be used to recognize and classify different website traffic despite attempts at encryption or tunneling. In this paper, I experimentally prove this concept. I extract coarse features of network traffic such as bandwidth, packet length, inter-arrival time, average packets sent/received per second of five popular websites namely, gmail, skype, facebook, gmail video chat, and youtube traffic and train the system. After sufficient training, I run the system on randomly collected traffic samples and calculated the accuracy of classification. During my experiments, I also observed that each website has different traffic profiles with respect to different browsers. Hence one cannot achieve accuracy in classification of traffic samples by training the system using a set of features obtained from traffic profiles collected using one browser and applying the same set of features on the sample traffic profiles collected using a different browser even for the same website.

1 Introduction

Although a rich literature exists about securing the confidentiality and integrity of messages transferred through internet, very little research has been undertaken to prevent any leakage of information. In order to enable secure transmission of data, internet users follow one of the following privacy enhancement technologies: establish an encrypted tunnel to a proxy and then relay the incoming and outgoing traffic through the tunnel or encrypt data by using WPA encryption protocol, if wireless network is used. Earlier, people believed that this type of encryption will protect them from malicious users in the internet.

The above mentioned privacy enhancment technologies offer protection against eavesdropping; however, it does not offer 100% protection of data. Monitoring the network traffic allows one to mount efficient traffic analysis attacks, despite the underlying data being encrypted. In general, the greater the quantity of packets being observed, or intercepted, the more information can be inferred from them. Public wifi network allows anybody (who has access to the access point) to intercept the data flowing in and out of the access point. Though the data packets are WPA encrypted, it is not safe from passive traffic analysis attacks.

These kind of attacks occurs because, though encryption provides confidentiality and authenticity of the data, it fails to hide coarse features of the traffic such as packet length, bandwidth, average packets transferred per second, the traffic burst, inter-arrival time of packets and its duration. With the help of these coarse features, one can effectively infer information about an interepted communication.

In order to mount any kind of attack, the attacker has to have access to some of the resources. In our scenario, the attacker has access to the access point, the traffic profiles of different websites(training data), different software tools to intercept the traffic sent and received to/from a given access point, the MAC address of the clients who are currently connected to the access point, and the encrypted packets sent and received by a client. With all this information, attacker can passively monitor the traffic of a target client and can mount an attack accordingly.

In this paper, I experimentally collected the traffic profiles, by visiting different websites and extracted the features that gave a consistent pattern across the collected traffic profiles of the same website. I used machine learning classifiers to train and test my system . To check the consistency of traffic profiles across web browsers, I collected traffic profiles of a website in Chrome and Mozilla and compared their traffic patterns. From these experiments, I observed that different web browsers exhibits varying patterns in traffic profiles for the same website.

2 Related Work

A wide range of research has been conducted in the field of traffic analysis. In 2002, Hintz [4] proposed the first passive traffic analysis attack in SafeWeb, an encrypting web proxy. He determined the identity of certain websites by looking at the total number of different packet lengths, number of connections made to the server, and the transfer of file sizes in each connection. These file sizes, however, could not be observed in encrypted tunnels.

Bissias et. al. [1] examined the distribution of packet sizes and inter-arrival time to classify websites from encrypted tunnels. Our experiments, indicate that these features might not be sufficient to distinguish the websites. The accuracy reported in [1] was around the 40% mark, which is not quite enough to realize accurate classification. In [3], Herrmann et. al presented a new classifier with a 97% reported accuracy. The classifier operated on packet size frequency distribution. However, extraction of a good feature set will lead to improved classification merely with basic classifiers like Naive Bayes and Decision tree method.

In recent work [2], the authors clearly elaborated the reason for the failure of all efficient traffic analysis countermeasures in preventing passive traffic analysis attacks. In this paper, I analysed the network traffic experimentally to understand the reason for the failure of those countermeasures. I collected profiles from different websites and tried to classify them using basic machine learning classifiers. Though I did this on a small scale, our system is easily scalable. With more efficient feature set, I can classify traffic profiles with 100% accuracy.

3 Methodology

In this section, I describe the methodology for performing the experiments. In the experiments, five popular, most frequently visited websites by internet users namely gmail, facebook, skype, youtube, and gmail video chat were considered. Website Fingerprinting involves three simple phases: data collection, feature extraction, training and testing.

3.1 Data Collection Phase

I visited each website separately, surfed the page for a given amount of time as a normal user would do and collected WPA encrypted traffic. I used airodump-ng for collecting WPA encrypted traffic. *Airodump-ng* is a packet sniffer for capturing raw 802.11 frames. It also writes out packets sent and received by access points and clients into a pcap file, which can then be analysed using *Wireshark*, a free and open-source packet analyzer used for network troubleshooting, analysis, software and communications protocol development, and education. Using Wireshark one can filter out unnecessary traffic, and get the traffic sent from the targeted host to the access point and from the access point to the targeted host.

3.2 Feature Extraction phase

Once sufficient data has been collected from different websites, I extracted features from the collected traffic profiles. From now on, I will refer to this as training data. Based on trial and error method, I came up with the following feature set.

Packet Length: Packet length is a variable, with certain websites transmitting data with a particular set of packet lengths. For e.g., From the experimental results, I observed that youtube traces got most of its packet lengths as 1500, 102, 137 or 1476, which are depicted in Figure 1. Similarly, gmail video chat got most of its packet lengths as 137, 102 or 1550, as shown in Figure 2.

Interarrival time between packets: The time difference between the previous packet and the next packet is called the inter-arrival time. Sometimes, inter-arrival time between packets will also differ among different websites, although in our experiments, I did not find any significant difference between the interarrival packet timings among different websites.

Upstream Bandwidth: Upstream is how fast a host(user) sends data to the particular web server in the network. It significantly varies among different websites.

Downstream Bandwidth: Downstream is how fast a host(user)receives data from the particular web server

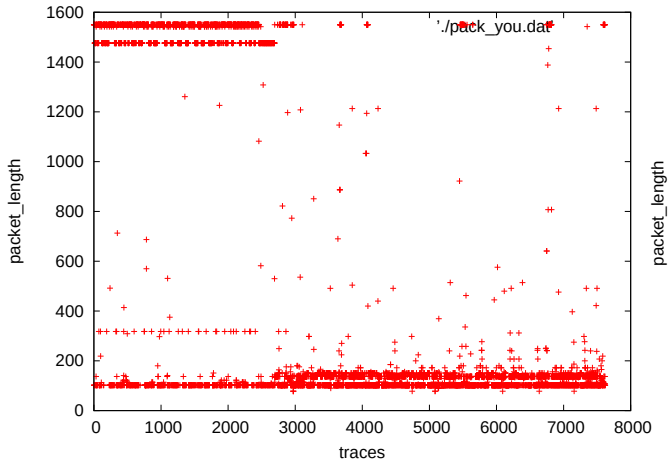

Fig. 1. Packet length distribution of youtube traffic

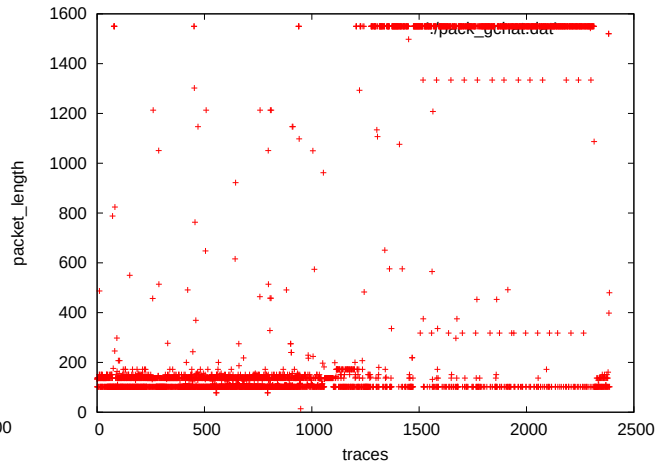

Fig. 2. Packet length distribution of gmail video chat

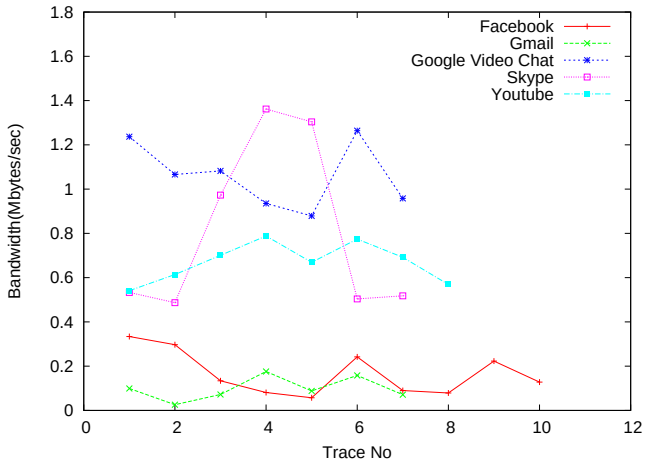

Fig. 3. Bandwidth distribution of various websites

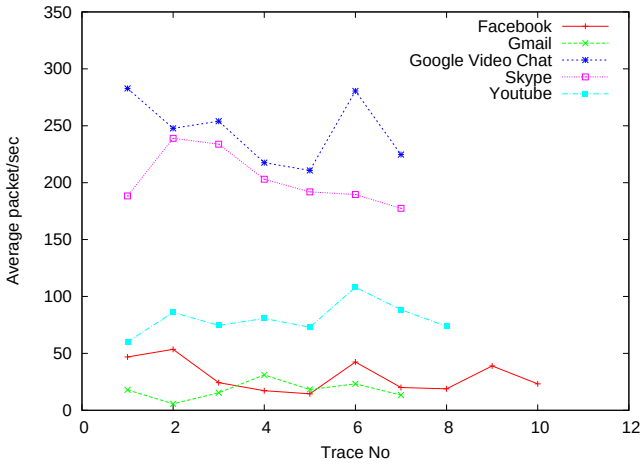

Fig. 4. Distribution of average packets transferred per second for various websites

in the network. Similar to upstream bandwidth, it also significantly differs among different websites. Figure 3 shows the total bandwidth distribution of different websites.

Average packets sent per second: This is defined as the amount of bytes transferred per second from the client to the web server for a particular website. For different websites, average packets sent per second, fall within different ranges.

Average packets received per second: Likewise, average packets received per second is defined as the amount of bytes transferred per second from the web server to a client. For a particular website, this value falls within a certain range. The distribution of average packets transferred per second for different websites is depicted in Fig 5.

3.3 Training and Testing

After obtaining sufficient traces for each of the different websites, I extracted the above mentioned features for each and every trace. I provided training to my system using Weka classifiers. I considered using Naive Bayes Simple, Naive Bayes and Decision tree based classifiers. Providing training using Weka is a very simple task which involves uploading the feature set file, selecting a classifier and finally clicking *Start* button. I then tested my system using randomly collected traffic profiles. The results are presented in the next section.

Our experiments show that the same set of features collected from one browser cannot be used as a training set for testing a different traffic profile obtained from a different browser. This is because, from my experiments, I found that the traffic profiles of the same website exhibit different characteristics across different browsers. Below are the graphs showing the variation in traffic characteristics, namely, bandwidth and average packets transferred per second between mozilla firefox and google chrome.

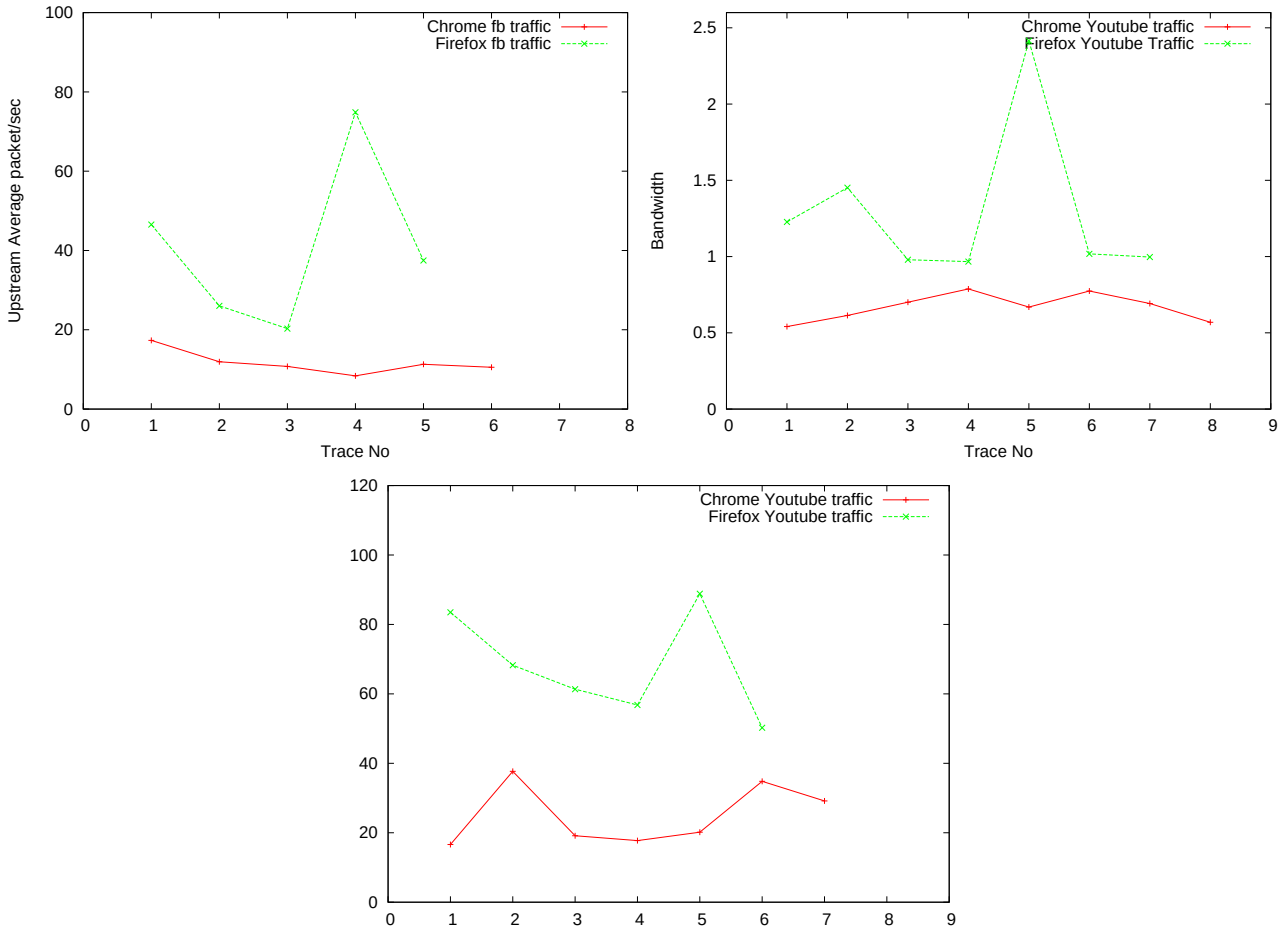

Fig. 5. Top Left: Distribution of average packet/sec for Chrome and Firefox facebook traffic Top Right: Bandwidth distribution for Chrome and Firefox youtube traffic distribution Bottom Center: Distribution of average packet/sec for Chrome and Firefox youtube traffic

Hence, I must train the system with traffic profiles obtained from different browsers. I can then then apply browser fingerprinting strategy [5], to find which browser the user is using and can mount my traffic analysis attacks accordingly.

4 Evaluation

The accuracy of classification obtained by using different classifiers are as follows:

Classifier	Facebook	Youtube	Gmail	Skype	Gmail Video Chat
Naive Bayes Simple	90%	100%	83%	89%	99%
Naive Bayes (without Supervised Discretion)	89%	100%	90%	89%	99%
Naive Bayes (with Supervised Discretion)	99%	100%	99%	100%	100%
Decision Tree(Rankers Search)	80%	100%	80%	90%	90%
Decision Tree(Best First Search)	100%	100%	100%	100%	100%

5 Limitation and Futurework

In this paper, I don't consider interleaved traffic, where the user can visit multiple sites simultaneously. I assume that the user will visit only one site at a time. But in real time scenario, this is not the case. The users can visit multiple websites at the same time. This is a major limitation of my system.

So far, I have tried classifying only five popular websites. This approach can be extended to many websites by providing appropriate training to my system. New features such as maximum frequency of a particular packet length can also be considered for achieving better accuracy.

6 Conclusion

In this paper, I experimentally proved that passive traffic analysis attacks are always possible, even though the messages are encrypted, as the countermeasures for such attacks do not hide the coarse features of the traffic. With the current method, I found that bandwidth and average packets transferred per second of websites such as gmail and facebook falls in the same range. Also, I observed that the website displaying video content has the same range of bandwidth. As the number of sites for classification increases, accuracy of classification may deteriorate. However, it should still be possible to differentiate between whether the user is watching a video or checking an e-mail.

References

1. George Dean Bissias, Marc Liberatore, David Jensen, and Brian Neil Levine. Privacy vulnerabilities in encrypted http streams. In *Privacy Enhancing Technologies*, pages 1–11, 2005.
2. Kevin P. Dyer, Scott E. Coull, Thomas Ristenpart, and Thomas Shrimpton. Peek-a-boo, i still see you: Why efficient traffic analysis countermeasures fail. In *IEEE Symposium on Security and Privacy*, pages 332–346, 2012.
3. Dominik Herrmann, Rolf Wendolsky, and Hannes Federrath. Website fingerprinting: attacking popular privacy enhancing technologies with the multinomial naïve-bayes classifier. In *CCSW*, pages 31–42, 2009.
4. Andrew Hintz. Fingerprinting websites using traffic analysis. In *Privacy Enhancing Technologies*, pages 171–178, 2002.
5. Ting-Fang Yen, Xin Huang, Fabian Monrose, and Michael K. Reiter. Browser fingerprinting from coarse traffic summaries: Techniques and implications. In *DIMVA*, pages 157–175, 2009.