Fundamentals of Computer Networks- CSE 534

Project Report
on
User Profiling in the Time of HTTPS

Submitted By:

Ratan Singh Ahirwar (112749385)

Anuroop Katiyar (112609023)

1. INTRODUCTION

Introduction of HTTPS was expected to make user profiling harder over the web. User profiling is basically a process by which we can identify the data about a user's interest. Such information is used by various third-party companies like Ad-agencies, search engines, etc. that target their audience on the basis of user's interest. In the case of HTTP there is no mechanism to secure the data flowing through the network and one can easily sniff data over a user connection and create user profiles. Thus HTTPS, an extension of HTTP, was implemented for secure data communication over a network.

In our project we try to show that even though HTTPS exists, we can still eavesdrop over an encrypted connection between the user and HTTPS webserver and get sufficient information to predict a user profile using traffic fingerprinting. We show that Server Name Indication (SNI) extension of TLS protocol provides the domain information that the user is visiting in an unencrypted format. In case of websites with similar content (like www.espnsports.com) across its pages, SNI can tell a lot about user's interests. For websites that have a variety of content, SNI information may not be sufficient to tell about a user's interest. For example, www.ebay.com does not tell us much about the user, but www.ebay.com/books/ would tell us that the user is interested in books, thus we can accurately build a user profile.

2. BACKGROUND AND MODEL

User profiling: Profiling systems use a closed source mapping between URLs and interest categories of the user. Building an accurate user profile, helps organizations to target their users with specific content that the users might be interested to have a look at. For our model we have chosen 25 websites across 5 different categories from Alexa Top Ranked Web Pages.

HTTPS and Server Name Indication: Hypertext Transfer Protocol Secure (HTTPS) is an extension of the HTTP. It is used for secure communication over a computer network and is widely used on the Internet. In HTTPS, the communication protocol is encrypted using Transport Layer Security (TLS) which provides a secure pipe for data transfer authenticated using a certificate. This secure pipe is established using a TLS handshake in which the client and server establish cryptographic keys to encrypt and authenticate data being transferred through the pipe.

Server Name Indication (SNI) is an extension to the TLS protocol by which a client indicated the hostname which it is attempting to connect to at the start of TLS handshake. This allows a server to present multiple certificates on the same IP address and hence allows multiple HTTPS websites to be hosted on the same server without the requirement of the different websites to use the same certificate. The hostname is not encrypted in the original SNI extension, so an eavesdropper can easily see which site the user is requesting. The hostname information is specified in the client_hello message, which is the first message of the TLS handshake.

Traffic Fingerprinting: Traffic fingerprinting is a technique used to sniff the web traffic by analyzing the data packets' flow pattern at the network/transport level without removing the encryption. Fingerprinting involves a training phase during which we build a fingerprint of each of the web pages being monitored. We can accomplish this by fetching the web pages multiple times and recording the features of the generated traffic such as packet size, number of packets, etc. Then we can eavesdrop on client's connection and extract same features from the traffic and try to match this trace to one of the fingerprints computed during the training phase. To build an accurate web fingerprinting we use features such as the size and direction of each packet of a TCP connection. Therefore, our classifier is robust against differences in bandwidth or congestion in route. Fingerprinting is hard in cases of an open-world scenario in which the client can browse any page outside the set of monitored web pages. In our project we show that webpage fingerprinting can be reasonably accurate in a closed-world scenario in which the eavesdropper monitors all the pages that the client can possibly visit. This assumption is realistic in our scenario because the eavesdropper knows the website requested by the user.

System Model: In our model we consider a situation where a network eavesdropper tries to profile users by sniffing on their network connections. We assume all connections to be over HTTPS, therefore the network eavesdropper does not have direct access to unencrypted data traffic being exchanged between the user browser and webservers. However, the eavesdropper can obtain the hostname requested by the user by looking at the SNI in the client_hello message. In cases where SNI is not used, client queries to DNS which can be used to obtain the hostname details.

For our model we simplify the structure of the website and user browsing behavior. For each website we only take the main page and the set of 1st level pages i.e. the pages linked to the main page. We do not consider any page beyond the ones linked to the main page, but our results can be generalized to account for complex website structures. We assume that a user visits one page at a time for each domain which could be the main page or any of the 1st level pages. The eavesdropper then tries to infer the page visited by the user and assign a particular category to the user's profile.

3. PROBLEM STATEMENT

The problem at hand is to design a methodology that can predict a user's profile using SNI information. For cases where the SNI information is not sufficient to predict a user's profile, we can use SVM (Support Vector Machine) to create a classifier for each of the monitored domains. This classifier can then be used to predict a mapping of the exact page the user is visiting from our trained data set. If we are able to build such a classifier with a decent prediction accuracy, we can then conclude that though HTTPS makes it difficult for profiling in general, but it does not completely eradicate it.

4. DATA COLLECTION

We first took 25 websites across 5 different categories – Business, Health, News, Shopping and Travel from the list of Alexa Top Ranked websites. For each website, we took approximately 15 URLs for Top Level Pages, resulting in a total of unique 364 URLs. Now for each URL we collected TCP dump over 2-3 days for 5 times, giving us a total of 1820 (5x364) dumps which were used as a training data set for our classifier. This dataset was used to train a classifier for an open world case i.e. when the trained data set does not contain entire details of the domains. For 2 websites – www.cnn.com and www.ebay.com, we took the complete list of top-level web page URLs (Total-259) and took the TCP dump 5 times for each URL. This data set was used to train a classifier for each of the domain in the closed world case.

5. SOLUTION

We now use the data collected to generate the following features for all the URLs:

a) Number of incoming packets
b) Size of each incoming packet
c) Number of outgoing packets
d) Size of each outgoing packet
e) 100 interpolation points from the cumulative in-packet size list

We use the above 104 features to train our classifier for the collected TCP dumps using SVM. Further we categorize our tests into below 2 categories:

1. **Open World:**
   a. We trained the classifier using 1820 dumps of webpages from various categories.
   b. Now we tried to match our trained data fingerprints with some test dumps.
   c. As this is an open-world scenario we took the test cases randomly which contained web pages outside the monitored data sets. Therefore, the prediction accuracy was very low in this case.

2. **Closed World:**
   a. We trained the classifier with top level pages of www.cnn.com and www.ebay.com based on the assumption that we already know the domain the user is trying to visit using SNI.
   b. Now we tried to match our trained data fingerprints with 25 test cases for each domain.
   c. Since this is a closed-world scenario, we have separate classifier for both the domains and trained dataset of all the top-level pages for each domain. Therefore, the prediction accuracy improved to 72% and 84% for ebay.com and cnn.com respectively.

6. EVALUATION AND RESULTS

We use an SVM classifier with an RBF kernel for different gamma values [ranging between 0.0001 to 10] and C values [ranging between 0.001 to 1000]. Now we plot a heatmap for tests across both the categories of open-world and closed-world.

Fig 1 below depicts no accuracy for C values in the range 0-0.5 in case of an open-world scenario. For C values in the range of 0.5-100 we get a low prediction accuracy of less than 10% for all gamma values.

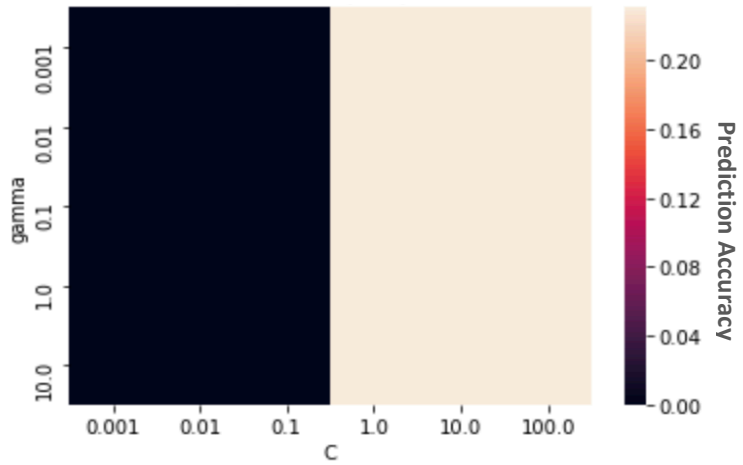

Fig 1 – Accuracy Heatmap for an open-world scenario

Fig 2 below depicts high accuracy for different gamma value ranges for cnn.com. The accuracy increases from 80% till around 90% when the gamma value is reduced from 10.0 to 0.0001.

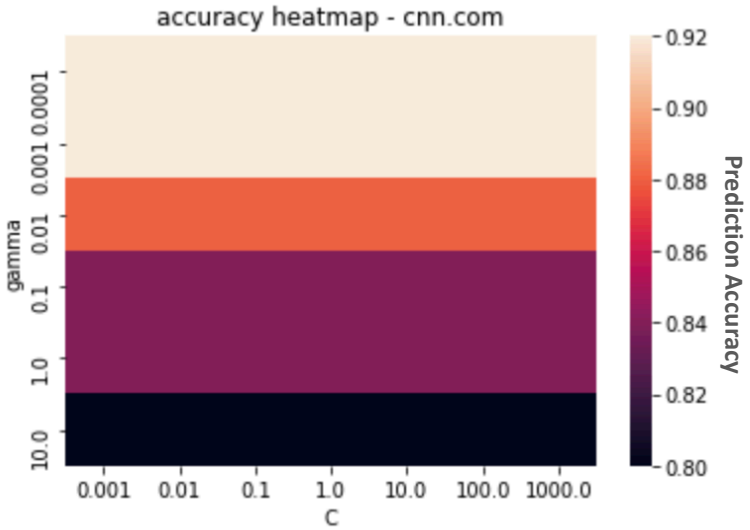

Fig 2 – Accuracy Heatmap for a closed-world scenario (www.cnn.com)

Fig 3 below depicts high accuracy for different gamma value ranges for ebay.com. There is no accuracy when C value is in range of 0.001 to 0.05. Prediction accuracy increases to nearly 80% for C value between 0.5 to 1000.0 and gamma range from 0.0001 to 10.0

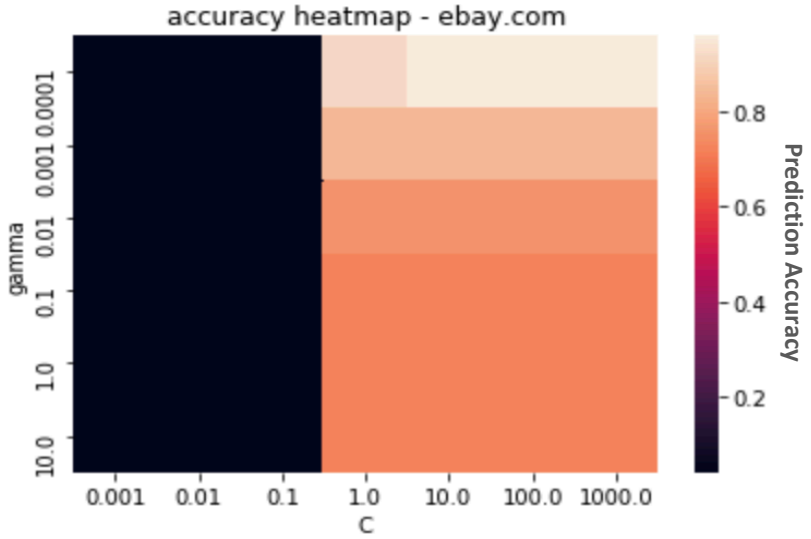

Fig 3 – Accuracy Heatmap for a closed-world scenario (www.ebay.com)

7. CONCLUSION

From the above results we can conclude that user profiling is achievable despite HTTPS if we can fingerprint the websites under observation. We have successfully shown that we can predict a user profile with high accuracy percentage in case of a closed-world scenario. Thus, HTTPS while being a tool to strengthen security of users over the web, cannot protect them against online profiling.

8. REFERENCES

[1] Roberto Gonzalez, Claudio Soriente, and Nikolaos Laoutaris, "User Profilin in the time of HTTPS" in Proc. of IMC'16.

[2] A. Panchenko, L. Niessen, A. Zinnen, and T. Engel, "Website fingerprinting in onion routing based anonymization networks," in Proc. of ACM WPES'11.

[3] "Alexa Top Sites." http://www.alexa.com/topsites.

[4] Link to our code repository - https://github.com/anuroop63/User-profiling-over-HTTPS