Google Summer of Code Proposal: A Framework for Website Fingerprinting Countermeasures

Marc Juarez Miro (Marc.JuarezMiro@esat.kuleuven.be)

Abstract

I would like to work in a framework that provides the basis for the further development of a Website Fingerprinting countermeasure. Despite the relevance of website fingerprinting attacks and the existence of some possible countermeasures, there is no implementation in Tor yet. The goal would be to prototype a countermeasure in a Pluggable Transport that would provide the primitives for the development of more sophisticated protocols.

1. What project would you like to work on?

Website Fingerprinting allows a local, passive eavesdropper who has access to the link between the Onion Proxy and the Entry Guard to learn user’s browser activity by leveraging packet sequence information. This attack breaks the privacy expectations of Tor users and although there are several countermeasures proposed, none of them has been yet implemented.

Most of these countermeasures are based on link-padding. This consists in adding dummy cells to conceal features of the traffic that could be exploited by the adversary. Since Tor is oriented to low-latency applications like web browsing, delaying strategies are unacceptable and, therefore, they must necessarily be based on padding. Link-padding incurs in high bandwidth overheads, however the increasing bandwidth capacity available makes it a relevant feasible defence.

Website Fingerprinting is calling the attention of the research community and there are several ongoing studies that tackle this topic. From the Tor community itself, there is also an interest in solving the issue. For instance, there is a bug ticket opened by Mike Perry that suggests the implementation of a tunable padding defense based on two proposed countermeasures [1].

Both countermeasures have some building blocks in common (e.g., a control logic to adapt padding under some specific conditions and reduce the overhead). The core of the project would be to implement them in the most general way as possible so they can be useful for other link-padding applications. For example, to implement the basis for a statistical model in the onion proxy and the entry guard that generates the appropriate cover traffic in both directions, where ‘appropriate’ would be abstracted as it would be defined by each specific countermeasure.
The goal is to implement these building blocks in a Pluggable Transport (PT) because it is the easiest way to prototype padding protocols without making modifications in the core program. In Figure 1 we show how we would use obfsproxy to implement the PT and add padding between the client and the bridge (in red the link being padded, in blue what is already implemented, yellow is what would be implemented and gray is future work).

![Figure 1](image)

Note that in order to frustrate an adversary that has access to the first link (including the entry guard) the padding should be extended until the middle node, as it could be that he controls the entry guard. For that we would need to modify the core of Tor. These modifications are out of the scope of this project. However, in the future we do not rule out the possibility of proposing a non-GSoC project to implement them in the Tor program.

**Deliverables**

The project is divided in the deliverables listed next.

**Bootstrapping (until May 19th)**

During this period I would be reading the specifications of pluggable transports protocol and familiarizing with python Twisted and the obfsproxy framework. At the end of this date, I should have an insight on obfsproxy and some running code to build upon.

**Implementation of BuFLO (week of May 19th)**

BuFLO is one of the most basic padding countermeasures. It homogeneously add packets at a constant rate. It does not take into account any information about pages so it’s very inefficient. However, to the cost of high overhead, it provide very good privacy guarantees. More advanced countermeasures are built upon BuFLO (such as Tamaraw and CS-BuFLO) and it could be a good starting point for the project.

BuFLO is specified with the following three parameters:
• \( d \), determines the size of the packets which in our case will be the size of a Tor cell (512Bytes)

• \( \rho \) determines the rate or frequency (in milliseconds) at which the clients sends packets

• \( \tau \) determines the minimum amount of time (in milliseconds) for which we must send packets

An important point which was missed in the original implementation of BuFLO is to establish correctly the limits of padding. Despite the efficiency constraints, we need to cover the original trace completely. Otherwise, the uncovered part can contain enough information to be a vector of website fingerprinting [2].

Handshake of padding protocol (from May 26th to June 2nd)

The parameters of BuFLO can be negotiated during the connection handshaking with the bridge. Some other information like the estimated size of the webpage can be used to establish the limits of padding. The handshake can be used also in more sophisticated protocols. The idea is to configure the padding for a specific website in the beginning of the connection, reducing as much as possible intermediate messages. At the end of this period we should have implemented the handshaking protocol.

Adaptive padding (from June 2nd to June 23th)

A straight-forward generalization of BuFLO is to send the packets following some probability distribution instead of sending them homogeneously. The distribution and the parameters of the distribution could also be agreed during the handshake. After this period we would like to have implemented an adaptive padding countermeasure and the extended the handshake protocol to that. We would try to generalize the BuFLO modules to accommodate the primitives of the adaptive padding.

For this countermeasure we will need to add state to the padding. This means that we will keep histograms for the number of packets that are sent and received. The padding will self-adapt to make this histograms look like some specific distribution, such as a different webpage.

We can abstract the input to such adaptive padding defense and implement some of them. For example, the Damerau-Levenshtein distance has been shown a good way to measure traffic [4]. We can implement an adaptive padding that uses edit-distance of traffic traces to adapt the padding and minimize the overhead.

General padding protocol (from June 23rd to August 4th)

Some capabilities for link-padding have been already specified for Tor by Steven Murdoch [3]. This functionality is very useful for website fingerprinting defenses but more sophisticated protocols are required for countermeasures that try to adjust the bandwidth overhead.
A possible extension of these basic Tor link padding capabilities could be to specify the number of cells to send in response to a padding cell request. This would offer more flexibility to the protocol. For example, a possible application could be to send a cell for each web object and receive the padding in response.

Statistical Models for Traffic Generation (from August 4th to August 18th)

A smart countermeasure that looks for the minimum bandwidth overhead should find the minimum cover traffic needed to pad the link and “fool” the adversary’s classifier forcing it to get a false positive. However, if we always try to optimize the bandwidth consumption, a strategic adversary who knows that such a defense is in place would be available to deploy statistical attacks and reverse the mapping from the optimally covered trace back to the original one.

That is why countermeasures should use an appropriate probability distribution to select the cover traffic randomly. In the election of the distribution we would strive for a trade-off between the average overhead and the vulnerability to statistical analysis. After this period we should have implemented some parameterized statistical models in the pluggable transport following a given distribution on a set of websites.

Evaluation Tools (Optional)

It would be of great help for the research and development community to implement tools to evaluate the countermeasures in place in the Tor network. This way we could measure the real bandwidth and latency overheads introduced. For this deliverable we would implement the methods needed to measure the performance of this countermeasure and estimate their impact in the Tor network.

This is an optional deliverable in case we finish to implement all the stated above before the coding season.

2. Why do you want to work with The Tor Project in particular?

The NSA revelations have raised deep concerns in me. I think that the mass surveillance and traffic analysis systems deployed by the intelligence agencies threaten or at least limit the freedom of citizens that use Internet. As it has been shown in some of the slides disclosed by Snowden, Tor is the tool that provides strongest privacy guarantees and has been set as a target by the NSA. For this reason, I think that working for the Tor project is the best way to pursue my aspirations for a fairer Internet.

3. Tell us about your experiences in free software development environments.

We especially want to hear examples of how you have collaborated with others rather than just working on a project by yourself.

I have been collaborating in the development of FP Detectives (GitHub), a general purpose framework to conduct large scale web privacy studies. In that project I worked together with Gunes Acar, another PhD student. I really liked the experience in collaborative development and I
am looking forward to work in larger teams. I have some experience with control version systems like SVN and Github.

4. Will you be working full-time on the project for the summer, or will you have other commitments too (a second job, classes, etc)? If you won’t be available full-time, please explain, and list timing if you know them for other major deadlines (e.g. exams). Having other activities isn’t a deal-breaker, but we don’t want to be surprised.

   I have a contract with my university as a PhD student. I would rather be working in the project part-time or as a volunteer (non-GSoC project).

5. Will your project need more work and/or maintenance after the summer ends? What are the chances you will stick around and help out with that and other related projects?

   Since the project is a prototype, future countermeasure will be build upon that. My plan is to stick in the development of future countermeasures and collaborate with the results of my research. Also, since a final website fingerprinting countermeasure should protect against an adversary that controls the entry guard, the padding in this case has to be run until the middle node. For that, further modifications in the Tor program are required. That is why eventually the project may turn into a tor-core proposal.

6. What is your ideal approach to keeping everybody informed of your progress, problems, and questions over the course of the project? Said another way, how much of a "manager" will you need your mentor to be?

   My preferred channels of communication are the mailing list and the IRC channel, and I plan to use them to ask questions related to the project. I would like to stay in close touch with my mentor and the Pluggable Transports developers. For some types of discussions it is also possible to have Skype meetings. I plan to report to my mentor once a week about my progress. Also, I will use a version control system such as GitHub, so that my mentor can check the code at any time.

7. What school are you attending? What year are you, and what’s your major/degree/focus? If you’re part of a research group, which one?

   I am first-year Ph.D. student at COSIC (CComputer Security and Industrial Cryptography) in KU Leuven, Belgium. Within COSIC, I am part of the Privacy Technologies group and my main focus is the study of protocols resistant to traffic analysis. In particular, I am currently working in a project related to website fingerprinting.
8. How can we contact you to ask you further questions? Google doesn’t share your contact details with us automatically, so you should include that in your application. In addition, what's your IRC nickname? Interacting with us on IRC will help us get to know you, and help you get to know our community.

Email: Marc.JuarezMiro@esat.kuleuven.be
IRC nickname: mjuarezm on OFTC

I am already in contact with the dev community through the mailing list and I also log in the IRC channel regularly.

9. Is there anything else that we should know that will make us like your project more?

This project would be the first step for a development of a countermeasure to website fingerprinting. Website fingerprinting is a critical vulnerability of Tor for two main reasons: (i) it defeats the main privacy guarantee offered by Tor (sender-recipient unlinkability), and (ii) an attack can be deployed with moderate resources (only access to the local network or to control an entry guard is required).

References