Covertness benchmarking of Tor pluggable transports

An MPhil project proposal

Chongyang Shi (*cs940*), Christ's College

Project Supervisor: Prof. Ross Anderson

Abstract

Censorship-circumventing Tor network traffic can be obfuscated as random traffic or traffic of a different protocol through the use of pluggable transport (PT) protocols. Past research efforts in detecting obfuscated PT traffic have yielded several categories of traffic analysis techniques, with varying performance and target protocol suitabilities. Inspired by related research on attacking image-watermarking systems [1], this proposed project intends to develop a benchmarking tool for evaluating the covertness of PT protocols through combinations of analysis techniques under current research. A baseline covertness against traffic classification could be established for development of new PT protocols.

1 Introduction, approach and outcomes

Tor is a popular tool for anonymised and censorship-resistant network communications. While it is trivial for a network node in a privileged position to detect and block non-obfuscated Tor traffic [2, Tb. 6] in a process called *traffic classification*, Tor provides a set of *pluggable transport* (PT) tools which clients can use to conceal their connections to a Tor bridge node from such censors. An arms race between state-sponsored censors and PT developers in traffic obfuscation has been going on for many years [3].

Among pluggable transports, two classes of techniques currently exist to achieve obfuscation of the encrypted traffic: pseudo-random transformation and mimicry of other "legitimate" protocols. Techniques in the former class attempt to avoid traffic classification by transforming Tor traffic into pseudo-random data, while those in the latter class transform Tor traffic into the likes of various other protocols that will result in too much collateral damage for the censor to block. A number of tools have been developed in each class and deployed with Tor distributions, with pseudo-random transformation represented by ScrambleSuit [4] and Format-Transforming Encryption (FTE) [5], and mimicry represented by meek [6] and SkypeMorph [7].

Obfuscation techniques can generally be evaluated on two metrics: the transmission performance under obfuscation, and the covertness of obfuscated traffic in regular traffic when examined by a state censor. For the purpose of censorship-circumvention, the latter is usually of most concern. There has been a few independent covertness evaluations on the aforementioned tools [8] [9] [10] over recent years. This is however still a relatively niche field of research when compared with related fields such as cipher cryptanalysis and steganography attacks, whose methodologies and techniques could be adapted into use in this field.

Under current research efforts, three categories of attack techniques are used to detect obfuscated traffic: semantics-based attacks where behaviour of traffic is checked against expected behaviours of its protocol [9, Sec. VIII] [10, Sec. 4]; entropy-based attacks where entropy signatures of packet payloads can be established for regular and obfuscated traffic [8] [10, Sec. 5]; and machine learning-based attacks that can be effective against protocols resistant to two previous categories of attacks [10, Sec. 6], but with significant drawbacks in portability between network environments [11].

All categories of attacks observe features in traffic traces such as packet metadata and distribution. Each category of techniques have distinct superiorities and weaknesses in terms of computational cost, protocol coverage, and portability. All attacks share the same the goal of achieving a high true-positive rate (identifying obfuscated traffic traces) and a low false-positive rate (not misidentifying non-obfuscated traffic as obfuscated), both of which are desirable to a state censor. Combinations of attacks from different categories can be chosen with deliberate strategies to maximise detection performance, as demonstrated by Wang et al. [10, Sec. 5.2], when semantics-based attacks and entropy-based attacks were used in conjunction to produce a high positive identification rate.

Therefore, the primary objective of this proposed project is to produce a benchmarking tool encompassing adapted versions of the aforementioned detection techniques. The tool will be able to accept sample traffic traces of any PT in use with Tor, and perform varying combinations of traffic analysis techniques to evaluate the covertness of the PT protocol (perhaps with obfuscated traffic likelihood scoring on results from multiple techniques). There is the possibility of automating the selection of combinations. With varying thresholds on acceptable true-positive and false-negative rates, it would be possible to estimate whether a PT protocol is of required covertness standards, similar to that performed by StirMark [1] on image-watermarking systems.

Individual detection techniques covered by past research efforts will be studied during integration into the benchmarking tool. As design and implementation details of PT protocols may have changed during the past few years, traffic traces from PT protocols will be examined again to verify previous observations on their identifying features. Attempts will also be made to find new features.

Traffic traces used in this proposed study need to closely resemble real network conditions. While it is possible for me to generate sets of PT and non-PT traffic traces by recording my own internet usage, to procure traffic traces at a large scale similar to that used by Wang et. al. [10], human volunteers can be invited to browse the internet in a monitored environment, subject to ethical review approval. This especially applies to non-PT traffic required to assess false-positive detection rates.

2 Workplan

By consulting reference implementations of some detection techniques by Wang et. al. [10] [1], there is a reasonable level of confidence that individual detection techniques can effectively detect obfuscated traffic to some level of accuracy. In the process of developing a benchmarking tool encompassing these techniques, the project will be focused on reimplementing, adapting and improving these techniques, so that they could be easily applied to arbitrary traffic traces without a lengthy parameter adjustment process (especially in the case of machine learning methods, to avoid overfitting).

Another significant implementation challenge is obtaining the traffic traces required for testing the benchmarking tool. In general, two sets of network traffic traces are required: a large set of regular

[1] Implementations by Wang et. al.: https://github.com/liangw89/obfs-detection

user traffic captured from regular internet browsing without the use of Tor and PT (required for false-positive detection test), and a smaller set of obfuscated PT traffic (consists of traces by different PT protocols for true-positive detection test). While it is possible to obtain anonymised real traffic traces from the Center for Applied Internet Data Analysis (CAIDA) [2], application-layer contents have been stripped during the anonymisation process for privacy. Synthetic traffic generated by automated browsing can also result in significant flaws in detection strategy, as observed by Wang et. al. [10, Sec. 6.1]. Therefore the only option is to capture and store full traces of real web browsing traffic.

As discussed in the earlier section, ethical review approval will be required for volunteers to be invited to create real world traffic at scale. Volunteers will be informed that their internet traffic will be recorded and stored securely for the purpose of the proposed project (traffic traces after anonymisation may be made available for reproducible research in the future). Volunteers will also be asked not to log into internet accounts or access personal information during the process. Only statistical data will be reported in the project report. Prior to an approval for this to be conducted, I will be able to create smaller scales of both sets of traffic by capturing my own internet usage, allowing preliminary work to be conducted.

With these considerations in mind, the proposed work plan for the project is as followed, divided into two-week chunks within a 28-week project period:

- **During Michaelmas Term**: Submit a request for departmental ethical review on the proposed volunteer traffic generation process, along with other project documents.
- **Chunk 1**: Study the reference implementation and select a suitable programming language for implementing the benchmarking tool. Preliminarily determine a general strategy for combined detection.
- **Chunk 2**: Create a mechanism for reliably capturing network traffic traces with Wireshark, Tor (with PT), and potentially over VPN to a capturing server. Test the mechanism by capturing own network traffic.
- **Chunk 3**: Study traffic traces of each PT to verify observations by past research efforts and attempt to identify new PT traffic identifying features.
- **Chunks 4-6**: Implement individual techniques, and test them on their own with traffic traces from own network usage. Implementations of detection techniques may be tuned based on findings from tests. If ethical review approves volunteer traffic generation, advertise with incentives for volunteers.
- **Chunks 7-8**: Conduct the volunteer traffic generation during this work segment. Implement the combined detection strategy and test with both sets of traffic traces.
- **Chunks 9-10**: Evaluate combined detection performance and make adjustments to combined and individual techniques as necessary.
- **Chunks 11-13**: Project report write up (the literature review will likely have been written during the previous chunks).
- **Chunk 14**: Reserved for contingencies.

[2] CAIDA: https://www.caida.org/data/

Project work may be conducted faster than planned if no significant problems occur and when time permits, in which case additional detection methods may be explored, especially in the case of methods based on machine learning.

References

[1] F. A. Petitcolas *et al.*, "Attacks on copyright marking systems," in *International workshop on information hiding*. Springer, 1998, pp. 218–238.

[2] T. Bujlow *et al.*, "Independent comparison of popular dpi tools for traffic classification," *Computer Networks*, vol. 76, pp. 75–89, 2015.

[3] S. Khattak *et al.*, "Systemization of pluggable transports for censorship resistance," *arXiv:1412.7448*, 2014.

[4] P. Winter *et al.*, "Scramblesuit: A polymorphic network protocol to circumvent censorship," in *Proceedings of the 12th ACM workshop on Workshop on privacy in the electronic society*. ACM, 2013, pp. 213–224.

[5] K. P. Dyer *et al.*, "Protocol misidentification made easy with format-transforming encryption," in *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security*. ACM, 2013, pp. 61–72.

[6] D. Fifield *et al.*, "Blocking-resistant communication through domain fronting," *Proceedings on Privacy Enhancing Technologies*, vol. 2015, no. 2, pp. 46–64, 2015.

[7] H. Mohajeri Moghaddam *et al.*, "Skypemorph: Protocol obfuscation for tor bridges," in *Proceedings of the 2012 ACM conference on Computer and communications security*. ACM, 2012, pp. 97–108.

[8] Q. Tan *et al.*, "Towards measuring unobservability in anonymous communication systems," *Journal of Computer Research and Development*, vol. 52, no. 10, p. 2373, 2015.

[9] A. Houmansadr *et al.*, "The parrot is dead: Observing unobservable network communications," in *Security and Privacy (SP), 2013 IEEE Symposium on*. IEEE, 2013, pp. 65–79.

[10] L. Wang *et al.*, "Seeing through network-protocol obfuscation," in *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*. ACM, 2015, pp. 57–69.

[11] L. Dixon *et al.*, "Network traffic obfuscation and automated internet censorship," *IEEE Security & Privacy*, vol. 14, no. 6, pp. 43–53, 2016.