POSTER: Fingerprinting Tor Hidden Services

Asya Mitseva
University of Luxembourg
asya.mitseva@uni.lu

Andriy Panchenko
University of Luxembourg
andriy.panchenko@uni.lu

Fabian Lanze
Huf Secure Mobile GmbH
fabian@lanze.net

Martin Henze
RWTH Aachen University
henze@comsys.rwth-aachen.de

Klaus Wehrle
RWTH Aachen University
wehrle@comsys.rwth-aachen.de

Thomas Engel
University of Luxembourg
thomas.engel@uni.lu

ABSTRACT

The website fingerprinting attack aims to infer the content of encrypted and anonymized connections by analyzing patterns from the communication such as packet sizes, their order, and direction. Although recent study has shown that no existing fingerprinting method scales in Tor when applied in realistic settings, this does not consider the case of Tor hidden services. In this work, we propose a two-phase fingerprinting approach applied in the scope of Tor hidden services and explore its scalability. We show that the success of the only previously proposed fingerprinting attack against hidden services strongly depends on the Tor version used; i.e., it may be applicable to less than 1.5% of connections to hidden services due to its requirement for control of the first anonymization node. In contrast, in our method, the attacker needs merely to be somewhere on the link between the client and the first anonymization node and the attack can be mounted for any connection to a hidden service.

Keywords

Tor Hidden Services; Website Fingerprinting; Privacy

1. INTRODUCTION

Tor is the most popular anonymization network. Daily, millions of people use it to hide their IP addresses while communicating on the Internet. By encrypting the traffic in layers and routing it over (at least) three nodes: *entry*, *middle*, and *exit*, Tor ensures unlinkability between communication partners. For many people, in particular for those living in oppressive regimes, the use of Tor is the only way to freely access information, without fearing consequences, or to bypass censorship. Besides protecting clients' privacy, Tor also allows servers to operate anonymously by offering (location-)*hidden services* (HSs). Following a special connection establishment procedure [1], the client can connect to the HS without needing to know its public identity. Tor hidden services allow users, e.g., human right activists and whistle-blowers, to exercise freedom of speech by publishing and offering access to content without being pursued, arrested, or forced to shut down their services.

The website fingerprinting (WFP) attack is a special case of traffic analysis, where a local observer (one of the weakest attackers in the attacker model for Tor) aims to identify the content (i.e., the page visited) of encrypted and anonymized connections by analyzing patterns of communication. Although Tor hides the content and addresses of communication partners, it is not able to obscure the size, direction and timing of transferred packets. Exploiting this information, the adversary, e.g., an ISP, located on the link between the user and the first Tor node (i.e., entry node) can capture patterns (i.e., fingerprints) from the transmitted traffic and discover which page the user visited.

Even though previous works have proposed WFP attacks feasible in closed-world settings [5, 7], i.e., the pages that a user may visit are limited to a fixed number[1], recent study has shown that no existing fingerprinting method is effective when applied in realistic settings [5]. However, the authors of [5] do not analyze the case of Tor hidden services. Contrary to the enormous variety and huge universe size (i.e., at least 4.75 billion pages[2]) of the world wide web (WWW), there are fewer than 60,000 hidden services[3] and only a few thousand of them provide HTTP(S) access [2]. Hence, if the adversary is able to reliably distinguish a HS connection from a regular one, we argue that the WFP attack within HSs will be a serious threat for the HS clients due to the small universe size. In this paper, we present our preliminary work on the WFP attack on Tor hidden services.

Recently, Kwon et al. [4] proposed the first WFP attack against HSs and their clients. Assuming the adversary controls an entry node, the authors detect the presence of HS activity by observing circuit-level information. However, their attack works only if all circuits to a HS go through a single entry node. In this work, we show that a user may contact several entry nodes to establish a connection to a HS. In particular, strongly depending on the Tor version used, the approach by Kwon et al. may be applicable to less than 1.5% of connections to HSs. In contrast to their method, we propose a two-phase fingerprinting approach which does not require an entry node to be controlled and to see circuit-level information. In our attack, the adversary merely needs to

[1]High accuracy has been shown for more than 1,500 pages.
[2]http://www.worldwidewebsize.com/ for July 2016.
[3]https://metrics.torproject.org/ for July 2016.

CCS'16 October 24-28, 2016, Vienna, Austria

ACM ISBN 978-1-4503-4139-4/16/10.

DOI: http://dx.doi.org/10.1145/2976749.2989054

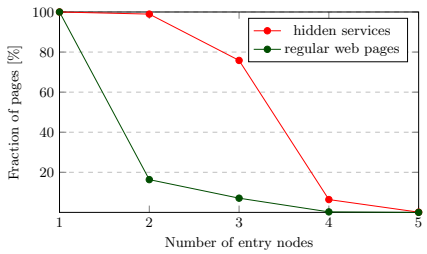

Figure 1: CCDF of the number of entry nodes used to load a HS vs. a regular web page.

be on the link between the client and the entry nodes. First, we attempt to detect a connection to a HS from the whole transmitted traffic. Once HS communication is detected, we try to determine the visited HS within the HS universe.

2. EXPERIMENTAL SETUP

To extract fingerprints (FPs) of HSs that provide HTTP(S) access, we first need to collect a representative set of onion addresses (i.e., URLs). To do this, we automatically crawled public search engines for HSs. To date, we collected 1,714 accessible onion addresses. As non-HS traffic, we randomly selected 20,000 unique pages from the `TOR-Exit` dataset [5] which represents pages actually accessed through Tor.

Since we assume that the adversary retrieves a certain amount of HSs and public web pages by himself as training data for fingerprinting, we applied the experimental setup presented in [5]. Using a toolbox containing the Tor Browser Bundle (TBB) 3.6.2 and *tcpdump*, we automatically recorded information such as size and direction of the TCP packets transferred during a page load. Although previous works have used TLS records and Tor cells to fingerprint pages [7, 5], recent study has shown that the difference in the detection rate among the separate extraction layers is negligible [5]. We then retrieved multiple traces for each page and removed faulty traces identifiable either by a Firefox connection error or by a HTTP status error code. We further excluded traces that have much lower or higher transmission size than the rest of traces related to the same page [5].

Observations. In contrast to previous work [4, 5, 7], where the authors assume that a page is loaded over one circuit, i.e., the clients communicate with only one entry node, we noticed that our clients typically connect to several entry nodes to fetch a single HS page. Figure 1 shows a CCDF of the number of entry nodes used to connect to a HS vs. a WWW page. While the client uses more than one entry node for fewer than 20% of WWW pages, almost every HS trace from our dataset is fetched over multiple entry nodes. Hence, in this case the WFP attack proposed by Kwon et al. [4] is applicable to fewer than 1.5% of connections to HSs.

The use of several entry nodes while connecting to a HS was also noticed by the Tor developers, who have introduced improvements in the newer TBB versions [6]. To evaluate these, we selected a sample of onion addresses from our dataset and collected traces for them by using the experimental setup described above, but applying TBB 5.5.5. As shown in Figure 2, the fraction of connections to HSs established over at least two entries decreased dramatically, i.e., down to fewer than 10%. As a consequence, contrary to common belief, at least with respect to HSs, the fingerprinting method of Kwon et al. has become dramatically more dangerous when recent TBB versions in place of earlier.

3. FINGERPRINTING APPROACH

To take advantage of the leak caused by HS connection establishment (i.e., the use of multiple entry nodes), we define the following features for our FPs. We first count the total number of distinct entry nodes used to load a single page. Next, we sum the packet sizes transmitted between the client and each of the entry nodes, denoted by S_{G_1}, S_{G_2}, S_{G_3}, S_{G_4} and S_{G_5}[4], ordered by size. If a given page is loaded over smaller number of entry nodes, the remaining values of these features, S_{G_i} for $2 \leq i \leq 5$, are zero. Finally, we include features representing the page load in the form of the chronological sequence of incoming and outgoing packets. To do this, we apply the fingerprinting technique proposed in [5]. Once the FPs are created, we use LibSVM [3] with a radial basis function kernel and apply 10-fold cross-validation.

4. HIDDEN SERVICE CLASSIFICATION

Our fingerprinting attack consists of two classification phases. In *phase one*, we try to detect a communication to a HS. Here, we further differentiate two scenarios, depending on whether the adversary is attempting to detect a communication establishment to an already-known or a new (i.e., not seen before) HS by using fingerprints of already-known hidden services. Once a HS communication is detected, in *phase two* we recognize the particular HS visited by a client. We leave the evaluation of phase two for future work.

Detection of unknown HS communication. To detect a communication to a HS, we consider *two-class* scenario where the whole set of HS pages forms a single class. Here, we call the set of HS pages, the *foreground* set, and the set of public web pages, the *background* set. For evaluation, we constructed the foreground set consisting of 1,714 HSs with one FP per HS, and the background set containing increasing sizes b of public web pages with one FP per page, with $b \in \{1000, 5000, 10000, 15000, 20000\}$.

In this phase, the accuracy, i.e., the probability of a true result (either true positive or true negative), cannot serve as indicator, since the sizes of the foreground and the background set are unbalanced. Therefore, we use two metrics commonly applied in similar domains: precision and recall. Recall corresponds to the probability that a communication to a HS is detected. Precision shows the probability that a classifier is actually correct in its decision when it claims to have detected a HS communication. Which metric is more important depends on the objective of the adversary. Since our primary goal here is to restrict the set of users to those that may have connected to a HS, recall is more important. However, from the attacker's perspective, both precision and recall should be ideally equal or close to one. In this case, he can be sure that all users connected to a HS are detected and the detection is practically always correct.

We calculated precision and recall for all values of b. As shown in Figure 3 and 4 (red line), although we observe a slow decrease of both metrics in general, they still remain high for increasing background set sizes. In particular, the users connected to HSs are detectable with a precision greater than 0.98 if the smallest universe is considered, and with a precision greater than 0.92 when the background set is extended to 20,000. We make similar observations for

[4]The number of distinct entry nodes used varies from one to five during our experiments.

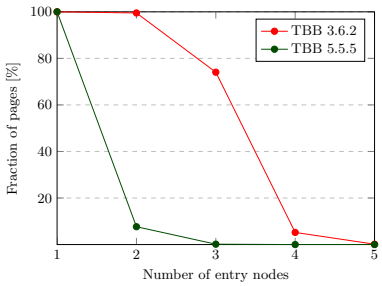

Figure 2: CCDF of the number of entry nodes for distinct TBBs.

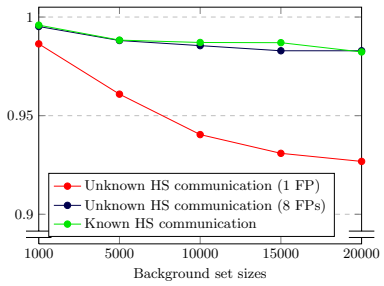

Figure 3: Precision for increasing background set sizes.

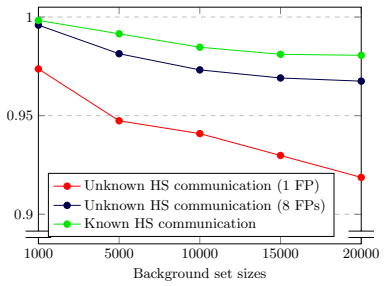

Figure 4: Recall for increasing background set sizes.

recall. The adversary is able to detect almost every HS client if b =1,000 and almost 0.92 of the HS connections for b =20,000.

To overcome the observed degradation in the results, we tried to increase the quantity of FPs per HS: instead of one, we applied eight FPs per HS, while keeping the background set as described above. The goal is to explore the number of FPs per a HS that the attacker needs for success. To ensure that all FPs belonging to a given HS are used only for training or for testing (but not for both), we applied an additional enclosing 10-fold cross-validation. Within each fold, we selected 90% of the data for training, i.e., for the foreground, 1,543 HSs with eight FPs per HS, and 10% of the data for testing, i.e., for the foreground, 171 HSs with eight FPs per HS, and calculated precision and recall for the all background set sizes. As shown in Figure 3 and 4 (blue line), for the different values of b, precision remains high, i.e., close to 1.0, and constant. A similar trend is observed for recall. For b =1,000, the adversary is able to detect every HS client, and almost 0.97 of the HS clients for b =20,000. Hence, with a moderate number of FPs per HS, the adversary is able to correctly recognize communication with an unknown HS.

Detection of known HS communication. So far we considered the case where the adversary tries to detect communication to a new HS that has not been seen before. However, due to the limited number of HSs it is feasible that the attacker knows of all available HS pages that a user visits. Hence, we want to study exposure to the WFP attack in the scenario where the attacker attempts to recognize known HS communication. To evaluate this, we extended our foreground set by one FP per HS, i.e., we considered 1,714 HSs with nine FPs per HS, while keeping the same background set. To apply 10-fold cross-validation, within each fold, we selected 90% of the HSs for training set, i.e., 1,543 HSs, with eight FPs per HS. For testing we utilized those FPs that are not included in the training set, but belong to the same HSs used to train the classifier. In total, 1,368 HS FPs were chosen to make the testing set equivalent to those applied for unknown HS communication. The background set was divided into training and testing set as above. We then calculated precision and recall for all values of b. As shown in Figure 3 and 4 (green line), both metrics are greater that 0.98 for all background set sizes, i.e., the classifier is able to detect almost every client connected to a HS and is always correct in its decision. Hence, detecting connection establishment to known HSs is an easier task. However, both known and unknown HSs can be detected with a high accuracy by using only a few thousand FPs for training.

5. CONCLUSION AND FUTURE WORK

In this work, we exposed the drawbacks of the currently existing method for fingerprinting HSs and showed that it may be applicable to fewer than 1.5% of connections. To overcome this, we presented a novel fingerprinting method which aims to detect connections to HSs without relying on malicious Tor nodes. Our approach is the first one that needs only a few collected traces per HS to correctly recognize almost every HS client. As next steps, we plan to estimate the scalability of our method by using larger background set sizes. We further plan to extend our HS dataset and evaluate phase two of our attack. We also plan to compare the performance of our approach with existing WFP attacks.

Ethical considerations. By completely gathering our set of onion addresses from public search engines, we do not harm HSs whose operators do not want to reveal their existence. We deployed our own clients to collect traces from the real Tor network and thus, prevented deanonymization of other Tor users. The ethical considerations for the `TOR-Exit` dataset discussed in [5] are also valid for this work. Concerning our future work and how to deal with HSs' sensitive information, we have already contacted the ethical board of the Tor research community.

Acknowledgements. Parts of this work have been funded by the EU H2020 project Privacy Flag and the Luxembourg National Research Fund (FNR) within the CORE Junior Track project PETIT.

6. REFERENCES

[1] Tor Rendezvous Specification. https://gitweb.torproject.org/torspec.git/tree/rend-spec.txt.

[2] A. Biryukov, I. Pustogarov, F. Thill, and R.-P. Weinmann. Content and popularity analysis of Tor hidden services. In *ICDCS*, 2014.

[3] C.-C. Chang and C.-J. Lin. LIBSVM: A Library for Support Vector Machines. *ACM TIST*, 2, 2011.

[4] A. Kwon, M. AlSabah, D. Lazar, M. Dacier, and S. Devadas. Circuit Fingerprinting Attacks: Passive Deanonymization of Tor Hidden Services. In *USENIX Security*, 2015.

[5] A. Panchenko, F. Lanze, A. Zinnen, M. Henze, J. Pennekamp, K. Wehrle, and T. Engel. Website Fingerprinting at Internet Scale. In *NDSS*, 2016.

[6] M. Perry. Notes and Action Items from Hidden Service Fingerprinting Session. https://lists.torproject.org/pipermail/tor-dev/2015-October/009632.html.

[7] T. Wang, X. Cai, R. Nithyanand, R. Johnson, and I. Goldberg. Effective Attacks and Provable Defenses for Website Fingerprinting. In *USENIX Security*, 2014.