Profiling Tor Users with Unsupervised Learning Techniques

Profiling Tor Users with Unsupervised Learning Techniques

Rafael Gálvez, Marc Juarez, and Claudia Diaz

KU Leuven, ESAT/COSIC and iMinds, Leuven, Belgium
{name.surname}@esat.kuleuven.be

Abstract. *Website fingerprinting* has been shown to be effective against Tor, one of the most popular low-latency anonymity networks. With this attack, a local network adversary is able to recover the browsing history of a client by using the traffic fingerprints observed at the client's connection to the Tor network. Previous studies on website fingerprinting focus on designing *supervised* classifiers to identify visits to a set of target websites. In this paper, we consider an adversary with the same capabilities as in website fingerprinting, but who uses unsupervised techniques to profile the users' browsing activity. We have used *OPTICS*, a clustering algorithm, to group similar traffic samples together, and the *BCubed Precision and Recall* metrics to measure the quality of the clustering. For a world of 100 websites, we show that, under mild assumptions, the attacker is able to group visits of different users to the same site with more than 50% success rate. We have also evaluated how the number of different pages that users can access impacts the effectiveness of the attack and found that for a world of 1,000 pages, the attack performance does not suffer a significant reduction.

1 Introduction

Website Fingerprinting (WF) is a traffic analysis attack that allows a network adversary to identify the websites that users are visiting by only observing their encrypted communications. The attack exploits differences in the timing and volume of traffic between different websites to create a fingerprint that can be matched to traffic traces generated by the user. WF has been shown effective in a wide variety of scenarios where Privacy Enhancing Technologies (PETs) were in place to protect the metadata of Internet traffic.

Among these technologies, Tor stands out as the most popular with two million daily users [27]. The Tor network is an overlay network that hides the metadata of the communication by using *onion routing* [11], which consists in wrapping messages with multiple layers of encryption. Each layer is encrypted with the key of a different node in the routing path, so that the message can be routed without any Tor relay knowing the origin and the destination at the same time. Tor is designed to protect against local network eavesdroppers that try to link the users with the pages they visit. However, some WF attacks in the

literature have proved highly effective in identifying the pages a user visits over Tor [5,28,13,23], thus breaking the anonymity properties of Tor.

WF is typically approached as a *supervised* learning problem, where websites are classification classes and each visit, characterized by its network traffic trace, is an observation or sample. The adversary first trains a classifier with samples of his own visits over Tor and then applies the classifier on user traces to guess the websites. To evaluate the classifier, prior work has made the assumption that the user can only access pages that the classifier has been trained on. This assumption is known as the "closed-world assumption" and is unrealistically favouring the adversary [17]. In such a closed-world scenario, the detection rates are high [28]; but expensive crawls, in terms of setup and maintenance, are needed in order to maintain high quality models [17].

A more challenging scenario is the "open world", where the user can visit any website even if the attacker has not trained the classifier on it. A recent study shows that the classifier accuracy drops dramatically in large open worlds [23]. In this paper, we shift the problem from a supervised to an unsupervised one: the attacker does not train a classifier to categorize the observations into web classes but clusters samples with *similar* properties. Thus, we consider an open-world by default and remove the need to maintain a database of website templates.

This unsupervised attack poses a more subtle threat to Tor user's privacy than traditional WF attacks. The type of inferences the attacker can make with clustering-based attacks are different from the ones in supervised WF. We show that such an attacker can effectively profile users based on the distribution of their visits, independently of the specific websites that are visited. For instance, an attacker that clusters traffic of multiple users can establish similarities in profiles of different users. This can complement previous WF attacks and aid target selection: if a users raises an alarm, an adversary may spend more resources on fingerprinting users with similar profiles.

In this paper, we explore the space of unsupervised techniques for profiling browsing activity and study the practicality of the attack for the use-case of Tor. In particular, the contributions of the following sections are:

Design an unsupervised profiling attack using clustering techniques. The features of our unsupervised attack are based on existing WF attacks. In Section 2, we review prior work on WF and describe the threat model considered throughout this paper. Then, in Section 3, we describe the features and the clustering algorithm on which we have based the attack.

Evaluate the effectiveness of the profiling attack in a realistic scenario. We describe the methodology for data collection and the dataset we have built for this study in Section 4. Next, Section 5 describes the metrics used to evaluate the success of the attack and Section 6 provides the results of the evaluation. We have evaluated the effectiveness of the attack in two different scenarios: a targeted attack, in which the attacker tries to profile one single user, and a non-targeted one, in which the attacker aims to profile multiple users. In the former, we show that the clustering made on data of a single user can be clustered accurately by the websites he visited and, in the latter, we demonstrate that the

attacker is able to link visits of different users to the same website. In Section 6 we also evaluate the impact of larger worlds of websites on the performance of the attack and found that the attack robustly withstands a world of 1,000 websites.

2 Adversary model and related work

In this section we define the adversary model considered in this paper and provide the necessary background on WF to understand and put in context the contributions of the rest of the paper.

As depicted in Figure 1, the threat model considered in most WF studies in Tor consists of a network adversary who is able to eavesdrop on the communication between a client and the entry *guard* to the Tor network. This adversary is assumed to be *passive*, since he can observe and record the traffic but cannot modify it; and *local*, as he cannot observe other parts of the network.

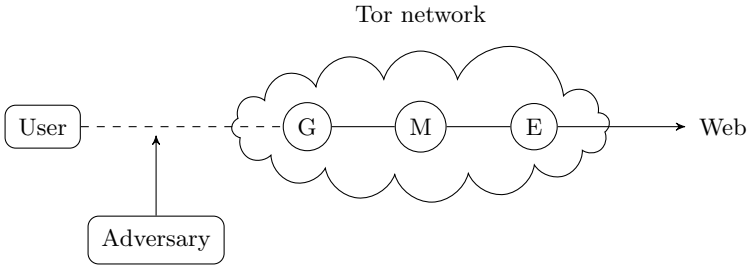

Fig. 1: The *User* browses the Web over a Tor circuit composed by three nodes: guard (G), middle (M), and exit (E). The *Adversary* eavesdrops the communication at the link between the user and the guard.

The adversary considered in this paper has the same capabilities as the attacker described above. However, his objectives are different from the ones of the typical WF attacker. While most of the work on WF studies attacks that recover the browsing history of a user, we consider an attacker that fingerprints websites to profile users according to their browser behaviour, independently of the specific websites they visit. Such an attacker wants to know whether the user has visited the same page multiple times, or whether two different users visit the same page regularly, regardless of the URL and location of these pages; this information can be used to deploy further attacks, more intensive in resources, including the existing supervised WF methods.

In WF studies, traffic traces are collected from visits to a list of pages and processed to serve as training instances for the classification model. The first WF classifiers were based on simple statistical models and were applied on SSL

traffic [6,22], web proxies [15] and, also, VPNs [26]. These studies assumed a *closed-world*, meaning that the user could only access a small set of websites. This assumption has been dismissed as unrealistic [17], as opposed to the *open-world*, where the user is allowed to visit any existing website.

In 2003, Liberatore and Levine presented the first attack using a machine learning technique for fingerprinting HTTPS traffic: a standard Naive Bayes classifier based on the unique lengths of network packets [20]. A few years later, Herrman et al. improved the classifier and applied it for the first time on Tor traffic [14], which only achieved 3% accuracy in a closed world of 775 pages, but encouraged other researchers to improve the attack.

Many attacks followed Herrmann et al.'s, using more complex models and refining the feature set: Panchenko et al. used an SVM with an euclidean distance based on burst-related traffic features [24]; Dyer et al. trained the Naive Bayes with Panchenko et al.'s features encoded as n-grams [8]; Cai et al. and Wang and Goldberg used SVMs with different edit-distances by representing traffic traces as strings [5,29]; Wang et al. presented a k-Nearest Neighbors classifier [28]; and, recently, Panchenko et al. have presented an attack that has proved the most successful attack to date [23], achieving over 90% accuracy in the closed world. This latter study concluded that, despite the high success rates achieved in the closed world, the WF attack does not scale to large open world scenarios.

Nevertheless, since the goal of the adversary we consider in this paper is not to identify specific pages, we do not require to train the classifier on labeled instances. In contrast to these works, our attacks are based on *clustering* techniques instead of classification. To the best of our knowledge, this is the first use of unsupervised learning for profiling Tor users based on their web browsing activity.

Although the implications of this attack are less obvious than in WF attacks, profiling users can be as threatening as the usual WF: it can reveal patterns in web browsing that compromise the privacy of Tor users. To illustrate this, imagine that the profile of a certain user stands out from the rest of users. This fact can uniquely identify the user and attract the attacker's attention, who may then mount more sophisticated attacks and distribute resources more efficiently. Another example is to cluster users that have similar profiles, so that learning information about a user reveals information about users with similar profiles.

3 Clustering Algorithms

The goal of the adversary in our threat model is to find patterns in a set of traffic samples, grouping those samples that are *similar*. Clustering algorithms are designed to tackle exactly this problem and the notion of similarity between samples strongly depends on the features that are selected to represent them [7]. In this section we discuss the election of the specific algorithm and feature set used to implement the clustering.

3.1 Features

In the WF literature, we find a broad range of feature sets that have been used for web traffic classification. All these features are built from the timing, direction and length of individual network packets in the traffic sample. Panchenko et al. and Dyer et al.'s work stands out for providing the first feature analysis for website fingerprinting and showed that coarse-grained features such as total transmission volume and traffic bursts are distinctive of a website [24,8].

For both, supervised and unsupervised learning, the features that provide best performance are those that make samples from the same site resemble each other, reducing the *intra-class variance*, and samples from different sites distinct to each other, increasing the *inter-class variance*. For this reason, we have chosen the feature set that gives best classification results in supervised learning, hoping that these properties will hold for unsupervised learning.

The features that give best results in WF are the ones in the state-of-the-art attack presented by Panchenko et al. [23]. They are the values of the function of instantaneous traffic volume, defined as the number of incoming bytes minus outgoing bytes, at a specific point in time. In order to extract the same number of features from this function, Panchenko et al. propose to always interpolate 100 equidistant points for each trace. We took the same parameters and used their implementation to extract these features from our dataset. The total number of incoming/outgoing bytes and packets in each trace are prepended to its feature vector adding up to a total of 104 features per vector.

We also chose these features because they are easy to understand and can be easily visualized and compared [23], so that even the human eye can assess the quality of a clustering using these features.

3.2 Algorithm

Density-based clustering algorithms group sets of samples that, according to a similarity metric, have higher density than other regions of the space. As opposed to more popular clustering algorithms such as K-means, these algorithms do not require the number of clusters as a parameter and try to infer it from the data itself. This property makes this type of algorithms specially well-suited for our attack, as the adversary does not know the number of classes and the number of samples for each class that he will observe in advance.

In particular, we picked OPTICS [3] from this family of algorithms. OPTICS is an extension of DBSCAN [10] that finds arbitrarily shaped clusters with varying densities. This is also an important property for our purposes, as the different update rates that websites have (e.g., news websites as opposed to web archive files) may induce different intra-cluster densities. We note that the goal of this paper is to demonstrate the practicability of our unsupervised attack and the exploration of the space of clustering algorithms is out of scope.

OPTICS takes two parameters, ε and $minPts$, and outputs a list of the samples in the dataset sorted by the distance of each sample to its $minPts$-nearest neighbors, as long as this distance is smaller than ε. Groups of points

that are either farther away than ε from any other point, or whose cardinality is less than $minPts$, are considered noise and are put into a special cluster, also called *rag-bag*. To extract the final clusters, we use a threshold on the distance between neighbours in the cluster, denoted as ε'. As before, we also impose the clusters to have at least $minPts$ samples.

There are several distance functions that can be defined in our vector space. The one proposed by Ankerst et al. in the original OPTICS paper is the Euclidean distance [3], but other distances may be used [12]. Based on a set of exploratory experiments in a small dataset, we selected three different distances:

- Euclidean: it is the most intuitive distance and gave good results in our first experiments. However, it is known to succumb to the so-called *curse of dimensionality* [1].
- Manhattan: this distance is also easy to visualize and is not as exposed to the curse of dimensionality as the Euclidean distance [1]. We assume that the reader is familiar with both Euclidean and Manhattan distances and we will skip the details.
- Shared Nearest Neighbors Jaccard Distance (SNN Jaccard): the SNN Jaccard between two samples takes the k nearest neighbors for each point and computes the Jaccard index between the two sets of neighbors [16].

ELKI [25] is an open source (AGPLv3) data mining software written in Java. The focus of ELKI is research on machine learning algorithms, with an emphasis on unsupervised methods for cluster analysis and outlier detection. ELKI is optimized to run and quickly evaluate clustering algorithms, which allows us to search the parameter space of the algorithm and find the arguments that give best results.

4 Experimental Setup

We based our data collection methodology on the one developed by Wang et al. [29]. For comparison with previous WF studies [5,29,28], we use the US Alexa list of top most popular domains. In particular, we used the top 100 from Juarez et al. [17], and top 1,000 most popular domains as per March 29, 2016.

Due to the dynamism of websites, traffic data tends to stale, making it more difficult to classify [17]. To mitigate the effect of staleness we crawled the Alexa top 100 in ten batches with four visits each. For the top 1,000, we visited each website 20 times in two batches of ten visits. The data collection started on March 14 and finished on May 7, 2016. At the time of our crawls, sixteen websites were showing a CAPTCHA to connections coming from the Tor network [19]. We discarded these traces along with a few others that were corrupt.

Contrary to what the name of the attack suggests, website fingerprinting studies consider attacks that fingerprint *webpages* and not websites. Virtually all studies on WF take only the home pages of the sites that they are trying to fingerprint. We will also make this assumption for this study and crawl only the home pages of the Alexa list.

To automate the visits to the webpages, we used the `tor-browser-crawler`[1], a Python module that drives the Tor Browser with `selenium`[2] to crawl a list of URLs. We used this crawler because it allows us to collect traffic using the same browser as regular Tor users. During our crawls we have contributed to the development of the crawler to support newer versions of the Tor Browser. The version used for our crawls was 5.5.5, the latest stable version.

There are a number of URLs in the Alexa list that map to localized versions of the same website (e.g., in the top 1,000 we find `google.com` and `google.de`. Some studies on WF have *localized* the Alexa list to only include only one version of the page, arguing that an attacker detecting a visit to `google.de` instead of `google.com` would learn roughly the same information. For our experiments we have mainly used the non-localized version of Alexa, as we did not observe a significant difference in attack performance compared to the localized version.

Following the recommendations of Elahi et al. [9], the Tor Project implemented a path selection algorithm that sticks to the same entry node for nine months. This is the default behavior in the current Tor Browser Bundle, so it is likely to be the most common setup among Tor clients.

In addition, we crawled the same list of sites with the `UseEntryGuards` flag disabled in the Tor configuration, instructing the client to pick a guard from all available entry nodes for every circuit. This is intended to simulate traffic generated by different users, as each visit is routed through a completely different circuit. We make the assumption that target users are located in the same network location and using similar devices (e.g., users in a university campus).

5 Evaluation

When measuring the quality of a clustering algorithm two kinds of criteria can be taken into account: *internal*, which describe the results based on metrics that evaluate intrinsic properties of the points in the clustering; and *external*, when the true classes of the points are available and can be used to assess the correctness of the result [21]. Internal criteria do not require available ground truth, but are agnostic to the specific problem that the clustering has been designed to solve. In this paper, the clustering algorithm is the basis of a profiling attack and the labels of the samples are available, thus external criteria are more appropriate for our evaluation. In particular, the metrics of our evaluation measure the following four properties of the resulting clusters [2]:

- Cluster homogeneity: the clusters must not mix traces belonging to different websites.
- Cluster completeness: traces belonging to the same website should fall into the same cluster.
- Rag bag cluster: one cluster should group all the noisy traces.

[1] https://github.com/webfp/tor-browser-crawler
[2] http://www.seleniumhq.org/

– Cluster size vs quantity of clusters: a small error in a big cluster should be preferable to a large number of small errors in small clusters.

Both cluster homogeneity and completeness are the basic constraints a good clustering algorithm must satisfy. They directly map to the *precision* and *recall* concepts from Information Retrieval theory [21]. A rag bag is useful in an open world scenario, since it limits the negative effects of "noisy" traces. OPTICS is tolerant to noise and has a rag bag of pages where noisy samples are discarded. The fourth property relates to the bias and variance trade-off, as it favours clusters that make a small error but generalize better in case we use the final clusters for classification.

5.1 BCubed Based Measures

There is only one single family of measures that satisfy the four properties above: the **BCubed based measures**. The BCubed based measures express the quality of the clustering in terms of Precision and Recall. The precision can be interpreted as the probability of picking two samples from a cluster that belong to the same website. The recall estimates the probability of picking two samples from different websites that fall into different clusters [2]. The former probability is informative of the success of the attacker if he picks n traces from the same cluster and assumes that belong to the same website. Conversely, the latter probability gives an estimate of the confidence of the attacker if assumes that two traces picked from different clusters belong to different websites.

In our evaluations, we have also used the F1-score measure, which is the harmonic mean of Precision and Recall and provides a combined metric: if either precision or recall are zero, the F1-score also is; and if one of them is high but the other is low, the F1-score is an average of the two factors.

6 Results

As introduced in Section 4, we take into account two different scenarios for the profiling attack: single and multiple guards. The metrics will be computed separately for both of them, using parameters guided by exploratory experiments on the Alexa top 100 dataset.

– Minimum number of points in a cluster ($minPts$): we set it between 0 and the maximum number of samples per website.
– The density threshold (ε'): it ranges between 0 and 1 and the best results were obtained with $\varepsilon' = 0.1$.
– Distance measure: as enumerated in 3, we evaluate Euclidean, Manhattan and SNN Jaccard distances.
– SNN Jaccard parameters: function that defines the neighborhood of a sample.

In addition to these parameters, we have normalized the dataset with the *min-max* scaling method which consists in normalizing the dataset by feature. We have noticed a significant improvement in the accuracy of the SVM used by Panchenko et al. [23] in the exploratory experiments with supervised classification. We also have noticed an improvement in the clustering after normalization. We note that this is a pre-processing step that can be applied by the attacker at any time as it only depends on the dataset itself.

6.1 Multiple Guards, 100 websites

In order to bootstrap the analysis of the 1,000 dataset, we use the top 100 dataset collected by Juarez et al. in 2014 [17] as a baseline to estimate the parameters described above.

Table 1 shows the value of our evaluation metrics for the best clusterings results with different distances.

Table 1: Precision and Recall for 100 websites.

Distance	Precision	Recall
Euclidean	0.51	0.61
Manhattan	0.53	0.59
SNN Jaccard	0.58	0.63

We can see that both the highest precision and recall values are attained using the SNN Jaccard distance, although the difference is smaller than ten percent points.

From the point of view of the attacker, these results imply that he can pick two samples from the same cluster that belong to the same website with probability of 0.58. If he selected two samples from two different websites, the probability of these samples falling into two different clusters is 0.63. These probabilities show that the profiling on one single user, although not highly accurate, approximates the actual websites browsed by the users. In the following section, we increase the number of websites that the user can visit to test whether this results also hold for the top 1,000 websites.

6.2 Comparison of Distance Metrics with 1,000 websites

In Figure 2 we show the precision and recall of the best clustering for each considered distance.

Notably, the Euclidean distance performs well despite the high dimensionality of the feature vectors (more than 100 features) in our unsupervised model. This is consistent with recent studies on machine learning that show that the curse of dimensionality does not affect large feature vectors if most of the dimensions

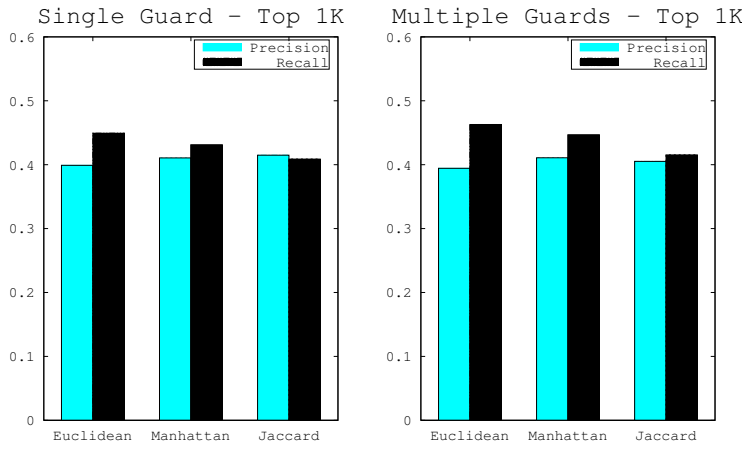

Fig. 2: Precision and Recall for Single and Multiple Guards top 1K. The Euclidean distance gives better recall than Manhattan and Jaccard, having lower precision.

are correlated [30], which is our case: most of the features are sampled from the volume of traffic in a given time, and its value at instant t_i depends on the volume at instant t_{i-1}).

6.3 Single and Multiple Guards, a Thousand Websites

We have repeated the clustering for the top 1,000 taking the same parameters as in the previous experiment and taking the Euclidean as a distance. Figure 3 shows the difference between the single and multiple guards datasets.

We would expect that pinning the guard would result in a reduction in the variance and would be an advantage for classification. However, we observe a slight increase in Recall in the multiple guard dataset compared to the single guard dataset. This difference is however very small and could be attributed to the gap in time between the single and the multiple guard crawls, or even to the higher heterogeneity of the data due to different locations of the entry guards. This would help the clustering algorithm to find more accurate limits to the clusters, since the difference between samples are higher.

The fact that the results observed in the multiple guard scenario are comparable with the one in single guard implies that the attack can cluster traffic samples of users in the same network location but using different guards, with a similar success than in the targeted attack.

Even though the 1,000 websites is a considerably large world compared to the closed worlds that have been considered in the WF literature, we are interested in studying how the clustering is affected by the size of the world.

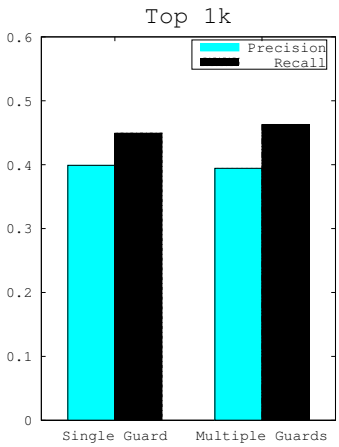

Fig. 3: Precision and Recall for the best clustering in the Single and Multiple Guards top 1K.

6.4 Different World Sizes

To test the evolution of the results depending on the world size, we clustered subsets of the top 1,000 dataset incrementally, from 200 to 1,000 websites in steps of a hundred websites. In Figure 4, we show the evolution of the best results for each world size. The graphs at the top and the bottom show the results for the single and multiple guard datasets, respectively.

In the single guard dataset the F1-score monotonically decreases, although the difference with respect to its value at 100 websites is negligible. In the multiple guard dataset, even though the F1-score is almost constant, we observe that Precision and Recall are actually fluctuating as the size of the world increases: recall is higher than precision for world sizes between 200 and 700, and the other way around elsewhere.

The size of the rag bag accounts for these variations in Precision and Recall in middle-sized worlds. We have seen that for the sizes of the world with higher precision than recall, the ratio between the number of instances in the rag bag over the total number of traces is relatively small. In those cases, the space of traces that do not fall in the rag bag becomes more sparse and OPTICS creates a greater number of smaller clusters, increasing the precision, because these clusters are more homogeneous, but decreasing the overall recall since it is easier for two samples from the same website to fall under different clusters. As the size of the world increases, the size of the rag bag levels off and the number of traces that potentially belong to a cluster outnumbers the noisy traces, that is why OPTICS is able to find good clusters again. We believe that this effect is due to the difference in intra-class variance between single and multiple guard.

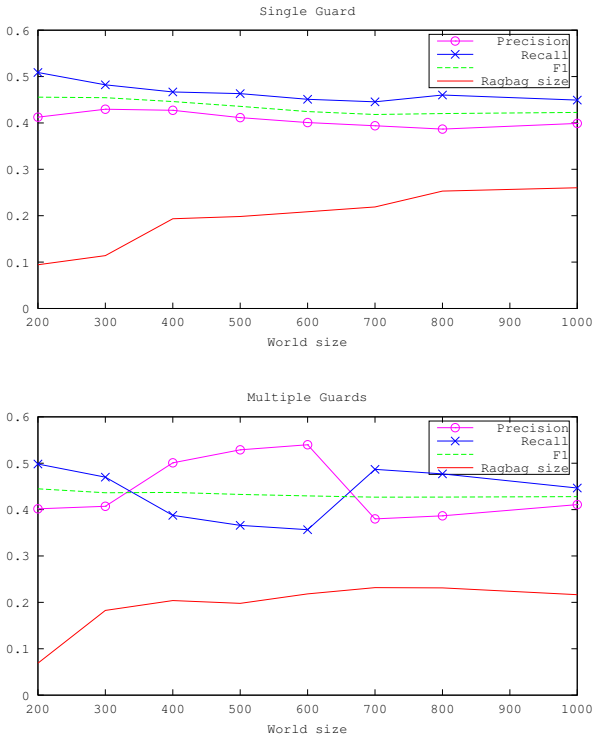

Fig. 4: The F1-Score, Precision and Recall for different sizes of the world. We also plot the size of the rag bag.

7 Discussion

While in classification the attacker can build a model out of observations of the objects that is trying to detect, clustering algorithms do not have explicit information about objects types and try to find patterns only based on their features. Therefore, the problem of clustering is considerably more challenging than the one tackled by supervised learning.

The adversary considered in this paper falls in Tor's threat model. The expectation of its users is Tor to decrease the confidence of the inferences that such an attacker can make on their web visits to random guessing. Given that the dataset has 20 samples per website, the probability of success of random guessing is approximately $(\frac{20}{1000})^2$, which is less than 0.0005%. However, this unsupervised attack allows to link the visits of two different users to the same website with almost 0.5 probability in average. Further, we tested the state-of-the-art WF

classifier and obtained an accuracy (recall) of 75% in the top 1,000 dataset, whereas the recall of the clustering algorithm is almost 50% using models whose average precision is 40%.

Furthermore, our Precision and Recall metrics take into account the rag bag as another cluster to evaluate. Ignoring it would give a biased estimation of the quality of the clustering. In practice, however, the attacker does not need to use the noise cluster to build profiles. If the number of noisy traces is sufficiently low with respect to the total number of traces, then the actual success of the attack would be higher than the reported by our measurements.

In regards to the applicability of this technique, it is worth noting that the results are comparable for both one user and a group of them, and that the parameters used to obtain the best clusterings for different world sizes are the same.

We have not evaluated any countermeasure to prevent this attack but, by the similarity of the features and learning objectives between WF and our unsupervised attack, we believe that WF defenses could effectively combat our attacks as well [4]. The lower performance of the unsupervised attack compared to WF suggests that lightweight defenses, such as the padding-based defenses proposed for WF [18], could at the same time mitigate the threat of unsupervised attacks.

8 Limitations and Future Work

Future work will address a number of limitations identified in this paper:

- The feature set was crafted for a supervised scenario. Thus, it is necessary to identify and obtain new features tailored to clustering algorithms.
- The dataset only contains traces of the home page. Visiting more pages from the same website would lead to a more realistic attack, and also would help to generalize it to bigger worlds by finding distinctive patterns between different pages from different websites.
- The evaluation algorithm promotes the creation of a cluster containing noisy traces. These may belong to a set of websites that share a broader category; following this example, an attacker could query if a set of people are visiting a sensitive categories such as "gambling" over Tor.
- The OPTICS parameter $minPts$ has been shown to remain constant across nine different world sizes, up to 1,000, with a prefixed ε. To generalize the validity of both $minPts$ and ε, it is necessary to test across more world sizes, as well as increasing the maximum number of websites.

9 Conclusion

In this paper, we have studied the practical feasibility of unsupervised techniques to profile the web browsing activities of Tor users. We have designed an unsupervised attack based on the OPTICS clustering algorithm and state-of-the-art WF features, and we have evaluated it in a number of scenarios. Our

results show that when trying to cluster visits of one single user to 100 different pages, the attacker has more than 50% Precision and Recall for all the distances we have considered in the evaluation.

We also have explored the application of the attack for profiling multiple users. We have simulated the traffic of multiple users by modifying the Tor configuration and have demonstrated that in a world of 100 websites, the attacker can link visits to the same website for different users with more than 50% confidence.

In addition, we evaluated the impact of the size of the world on the attack's performance. We have applied the attack under the same conditions on different numbers of websites up to 1,000. Our results show that the performance of the attack is not greatly impacted by the size of the world as far as we have observed in our experiments.

These results provide evidence that inferences on unlabelled Tor traffic data are possible, diminishing the privacy and security that Tor is expected to provide. This is the first work on unsupervised attacks for web profiling in the Tor network. Even though it still requires parameter tuning, we hope it will open the door to further research on this topic. Future work should explore more advanced unsupervised techniques and further investigate its practicability in the wild.

Acknowledgments

This work was partially by the Research Council KU Leuven: C16/15/058, by KU Leuven BOF OT/13/070, by the Flemish Government FWO G.0360.11N Location Privacy, FWO G.068611N Data mining and by the European Commission through H2020-DS-2014-653497 PANORAMIX and H2020-ICT-2014-644371 WITDOM. Marc Juarez is funded by a PhD fellowship of the Fund for Scientific Research - Flanders (FWO).

References

1. Aggarwal, C.C., Hinneburg, A., Keim, D.A.: On the surprising behavior of distance metrics in high dimensional space. Springer (2001)
2. Amigó, E., Gonzalo, J., Artiles, J., Verdejo, F.: A comparison of extrinsic clustering evaluation metrics based on formal constraints. Information Retrieval 12(4), 461–486 (2008)
3. Ankerst, M., Breunig, M.M., Kriegel, H.P., Sander, J.: OPTICS: ordering points to identify the clustering structure. In: ACM SIGMOD International Conference on Management of Data. pp. 49–60. ACM (1999)
4. Cai, X., Nithyanand, R., Johnson, R.: CS-BuFLO: A Congestion Sensitive Website Fingerprinting Defense. In: Workshop on Privacy in the Electronic Society (WPES). pp. 121–130. ACM (2014)
5. Cai, X., Zhang, X.C., Joshi, B., Johnson, R.: Touching from a distance: Website fingerprinting attacks and defenses. In: ACM Conference on Computer and Communications Security (CCS). pp. 605–616. CCS '12, ACM (2012)

6. Cheng, H., Avnur, R.: Traffic Analysis of SSL Encrypted Web Browsing. Project paper, University of Berkeley (1998), Available at http://www.cs.berkeley.edu/~daw/teaching/cs261-f98/projects/final-reports/ronathan-heyning.ps
7. Dy, J.G., Brodley, C.E.: Feature selection for unsupervised learning. J. Mach. Learn. Res. 5, 845–889 (Dec 2004)
8. Dyer, K.P., Coull, S.E., Ristenpart, T., Shrimpton, T.: Peek-a-boo, i still see you: Why efficient traffic analysis countermeasures fail. In: 2012 IEEE Symposium on Security and Privacy. pp. 332–346 (May 2012)
9. Elahi, T., Bauer, K., AlSabah, M., Dingledine, R., Goldberg, I.: Changing of the guards: a framework for understanding and improving entry guard selection in tor. In: Proceedings of the 2012 ACM Workshop on Privacy in the Electronic Society. pp. 43–54. WPES '12, ACM, New York, NY, USA (2012)
10. Ester, M., Kriegel, H.P., Sander, J., Xu, X.: A density-based algorithm for discovering clusters in large spatial databases with noise. In: Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD-96). pp. 226–231. AAAI Press (1996)
11. Goldschlag, D.M., Reed, M.G., Syverson, P.F.: International workshop on information hiding. In: Hiding Routing information. pp. 137–150. Springer Berlin Heidelberg, Berlin, Heidelberg (1996)
12. Grabusts, P.: The choice of metrics for clustering algorithms. Environment. Technology. Resources. Proceedings of the International Scientific and Practical Conference 2(0), 70–76 (2015)
13. Hayes, J., Danezis, G.: Website fingerprinting at scale. Tech. rep., University College of London (UCL) (2015), Technical report in arXiv.org.
14. Herrmann, D., Wendolsky, R., Federrath, H.: Website fingerprinting: Attacking popular privacy enhancing technologies with the multinomial naïve-bayes classifier. In: ACM Workshop on Cloud Computing Security. pp. 31–42. ACM, New York, NY, USA (2009)
15. Hintz, A.: Privacy enhancing technologies (pets). In: Fingerprinting Websites Using Traffic Analysis. pp. 171–178. Springer Berlin Heidelberg, Berlin, Heidelberg (2003)
16. Jarvis, R.A., Patrick, E.A.: Clustering using a similarity measure based on shared near neighbors. IEEE Transactions on Computers C-22(11), 1025–1034 (Nov 1973)
17. Juarez, M., Afroz, S., Acar, G., Diaz, C., Greenstadt, R.: A critical evaluation of website fingerprinting attacks. In: Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security. pp. 263–274. CCS '14, ACM, New York, NY, USA (2014)
18. Juarez, M., Imani, M., Perry, M., Diaz, C., Wright, M.: Wtf-pad: Toward an efficient website fingerprinting defense for tor. arXiv preprint (2015)
19. Khattak, S., Fifield, D., Afroz, S., Javed, M., Sundaresan, S., Paxson, V., Murdoch, S.J., McCoy, D.: Do you see what i see? differential treatment of anonymous users. In: Network and Distributed System Security Symposium (2016)
20. Liberatore, M., Levine, B.N.: Inferring the Source of Encrypted HTTP Connections. In: ACM Conference on Computer and Communications Security (CCS). pp. 255–263. ACM (2006)
21. Manning, C.D., Raghavan, P., Schütze, H., et al.: Introduction to information retrieval, vol. 1. Cambridge university press Cambridge (2008)
22. Mistry, S., Raman, B.: Quantifying Traffic Analysis of Encrypted Web-Browsing. Project paper, University of Berkeley (1998), Available at http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.10.5823&rep=rep1&type=pdf

23. Panchenko, A., Lanze, F., Zinnen, A., Henze, M., Pennekamp, J., Wehrle, K., Engel, T.: Website fingerprinting at internet scale. In: Network & Distributed System Security Symposium (NDSS). IEEE Computer Society (2016)
24. Panchenko, A., Niessen, L., Zinnen, A., Engel, T.: Website fingerprinting in onion routing based anonymization networks. In: ACM Workshop on Privacy in the Electronic Society (WPES). pp. 103–114. ACM (2011)
25. Schubert, E., Koos, A., Emrich, T., Züfle, A., Schmid, K.A., Zimek, A.: A framework for clustering uncertain data. PVLDB 8(12), 1976–1987 (2015)
26. Sun, Q., Simon, D.R., Wang, Y.M.: Statistical Identification of Encrypted Web Browsing Traffic. In: IEEE Symposium on Security and Privacy (S&P). pp. 19–30. IEEE (2002)
27. Tor project: Users statistics. `https://metrics.torproject.org/users.html`, (accessed: December 18, 2015)
28. Wang, T., Cai, X., Nithyanand, R., Johnson, R., Goldberg, I.: Effective attacks and provable defenses for website fingerprinting. In: USENIX Security Symposium (USENIX). pp. 143–157. USENIX Association (Aug 2014)
29. Wang, T., Goldberg, I.: Improved Website Fingerprinting on Tor. In: ACM Workshop on Privacy in the Electronic Society (WPES). pp. 201–212. ACM (2013)
30. Zimek, A., Schubert, E., Kriegel, H.P.: A survey on unsupervised outlier detection in high-dimensional numerical data. Statistical Analysis and Data Mining 5(5), 363–387 (2012)