The One-Page Setting: A Higher Standard for Evaluating Website Fingerprinting Defenses

ABSTRACT

To defeat Website Fingerprinting (WF) attacks that threaten privacy on anonymity technologies such as Tor, defenses have been proposed and evaluated under the multi-page setting. The multi-page setting was designed as a difficult setting for the attacker and therefore gives too much of an advantage to the defense, allowing weak defenses to show success. We argue that all WF defenses should instead be evaluated under the one-page setting so that the defender needs to meet a higher standard of success.

Evaluating known WF defenses under the one-page setting, we found that Decoy, Front and Tamaraw all failed to defend against WF attacks. None of these defenses were shown to be vulnerable in previous work. In Tamaraw's case, the attacker's TPR increases 13 times from 2.9% to 37% with 4.4% FPR; he can also achieve 91% TPR and 21% FPR. We also found that these attacks were able to succeed in a wide array of newly defined WF scenarios that could not be captured by the standard laboratory scenario. In response, we create the first defense that is strong enough for the one-page setting by augmenting Tamaraw with greater randomization overhead so that its anonymity sets are more evenly dispersed.

CCS CONCEPTS

• **Security and privacy → Pseudonymity, anonymity and untraceability**; • **Networks → Network privacy and anonymity**.

KEYWORDS

Anonymity networks; traffic analysis; website fingerprinting

1 INTRODUCTION

Internet users are constantly subjected to eavesdropping and surveillance. Anonymity networks, such as Tor, protect user privacy by relaying their traffic across multiple volunteer nodes, such that an eavesdropper at a single location cannot capture both their identity and their behavior simultaneously. However, Tor is vulnerable to traffic analysis attacks known as Website Fingerprinting (WF), which allow an eavesdropper (including Tor's volunteer nodes) to determine the user's activity from traffic patterns. WF attacks have been repeatedly shown to achieve high recall and/or precision in a large open-world setting [9, 11, 15, 17, 18].

On the other hand, there has been little success in work on WF defenses — mechanisms to obfuscate web traffic so that WF attackers cannot recognize them. As of yet, Tor does not use a single WF defense, despite more than a decade of research and implementation work. This is in large part due to the cat-and-mouse game that characterizes research on defenses. A defense would be published that shows success against all known attacks, but it is soon followed by a new attack that defeats this defense, necessitating work on a new defense. Tor developers are unlikely to invest the technical effort necessary to maintain a defense that may soon be defeated. Attempts have been made to break this cycle by designing provably effective WF defenses that can defeat any theoretical WF attack, but these defenses are prohibitively expensive in overhead and less suitable for use in a popular (and thus resource-strapped) anonymity network such as Tor.

In this work, we investigate the question: **How should we show that a WF defense is effective?** We find that defenses have been evaluated using a methodology that strongly favors the defender: it requires the attacker to distinguish between a large number of classes in the open world. This methodology was directly transposed from research work on attacks where it was designed to demonstrate attack effectiveness in a difficult setting. When used for defense evaluation, the same methodology only shows the success of a defense in an easy setting. In reality, an attacker may only need to recognize accesses to a single web page (which we call the **one-page** setting), and a defense must still thwart that attempt. The low bar for defense evaluation explains why historically, many defenses have been quickly beaten by newer attacks.

Our work re-evaluates WF defenses using the one-page setting with the following main contributions:

(1) In the one-page setting, we find that attacks can achieve high (>90%) TPR even against defenses that were not considered broken. We analyze defenses separately to explain how they fail to cover visits to a page, particularly highlighting limitations in current design paradigms.
(2) We newly define a number of realistic scenarios in which the WF attacker's success is not directly captured by the standard laboratory scenario. We demonstrate that a realistic WF attacker can achieve his goals even if the false positive rate is much higher than the base rate. We also reveal a number of important variables for attacker success that had previously been ignored.
(3) We attempt to fortify WF defenses in the one-page setting by exploring randomness and regularization options for several defenses. In doing so, we find that some defense paradigms have more potential to be fortified than others, and we are able to create the first defense that succeeds in the one-page setting (though with high overhead).

We organize the rest of the paper as follows. We give the background of Tor and Website Fingerprinting in Section 2, where we also explain the weaknesses of the previous evaluation methodology and our proposed one-page setting. In Section 3, we evaluate WF defenses in the one-page setting. We define realistic scenarios where an WF attacker can succeed against these defenses in Section 4. Then, we explore improvements to these defenses in Section 5. We discuss relevant issues in Section 6, give related work in Section 7, and conclude with potential future work in Section 8.

Anonymous Submission to ACM CCS '21, November 14–19, 2021,
2021. ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

2 BACKGROUND AND METHODOLOGY

2.1 Tor

Tor is a widely popular anonymity network designed for low-latency internet usage such as web browsing [6]. By relaying client traffic across multiple (volunteer) nodes with layered encryption, Tor ensures that only the entry node contacted by the user knows their identity, while only the exit node sees the end server. The separation of identity and activity safeguards privacy, and eavesdroppers on the network should not be able to link entry node traffic with the true web server being visited.

2.2 Website Fingerprinting Attacks

In Website Fingerprinting (WF), a local eavesdropper (which may include the entry node) uses traffic analysis techniques on the client's traffic to deduce the web page they are visiting, thus compromising Tor's main guarantee of privacy. The threat of such traffic analysis attacks against web privacy was studied before Tor [16], and was considered a potential threat at Tor's creation [6]. A large number of attacks have demonstrated success against Tor in a large multi-class open-world setting [1, 9, 11, 13, 15, 18]. Even in scenarios with very low base rates, WF attacks can achieve high precision with almost no false positives if a true positive rate trade off is acceptable [17].

We select three of the most effective attacks for evaluating defenses in the one-page setting:

- k-Fingerprinting (kFP) [9]: A classifier based on random forests, each forest being a multiple decision tree. The decision trees are trained on a large set of features such as packet counts, inter-arrival times and burst patterns.
- CUMUL [11]: An SVM trained on cumulative packet size sums, with outgoing packets from the client counting positively to the cumulative sum and incoming packets counting negatively. Notable for using a small number of features (104).
- Deep Fingerprinting (DF) [15]: A Convolutional Neural Network taking packet directions as input; it is currently the state-of-the-art attack.

While there may be other WF attacks based on deep learning [1, 13], these attacks are sufficient for our evaluation of defense performance.

2.3 Website Fingerprinting Defenses

To harden Tor against these attacks, researchers have proposed a number of WF defenses [2, 7, 8, 10, 12]. Broadly, defenses can be classified into one of three types:

- **Noise:** adding dummy packets in a random fashion to disrupt classification. Examples include Front [8], which adds dummy packets according to a Rayleigh distribution focusing on covering the front of the packet sequence, and WTF-PAD [10], which adds dummy packets to attempt to mimic a interpacket timing distribution.
- **Mimicry:** disguising the traffic of a page to look like that of another page. An example is Decoy [12] randomly loading a decoy web page whenever a real web page is loaded.
- **Regularization:** defining fixed rules and patterns for all web traffic to follow in order to limit feature leakage. An example is BuFLO [7] and the later improvement Tamaraw [2], which stipulate fixed packet rates that all packet sequences must follow, delaying real data and adding dummy packets as necessary, as well as only allowing a sequence to end at specified lengths to reduce leakage.

We focus on evaluating WTF-PAD, Front, Decoy, and Tamaraw in this work as representative defenses from each type. Currently, no WF attack has shown success against Front, Decoy, and Tamaraw.

2.4 Classification Basics

In the WF classification problem, the attacker (Oscar) obtains web traffic traces by passive eavesdropping, and attempts to classify them as positive (sensitive) or negative (non-sensitive) web page accesses. The attacker's goal determines what he considers sensitive and non-sensitive. We also refer to sensitive pages as monitored pages and non-sensitive pages as non-monitored pages. For positive web page accesses, the attacker also wants to identify exactly which page the client has visited. The problem to be solved by the attacker is a hybrid between multi-class classification and binary (sensitive/non-sensitive) classification, and we refer to it as the multi-page open-world problem.

The client/defender (Alice) may try to obfuscate these traces to thwart the attacker. Since the defender is aligned with the client, we do not distinguish between them. A network-layer defender can delay packets and insert new dummy packets at specific times. She may do so based on which page the client is truly visiting.

On a multi-hop anonymity network such as Tor, the attacker sits between the client and the first node. Since the first node can also be an attacker, dummy packets are dropped by the second node. Due to layered encryption, the attacker cannot read any packet, which also means he cannot identify which packets are dummy packets. He only knows the timing, size, and direction (to or from the client) of each packet.

The attacker's success is measured in his True Positive Rate (TPR) (or recall) and his False Positive Rate (FPR). It may also be measured in his precision, the percentage of positive classifications that are true, keeping in mind the base rate, which is the client's chance of visiting a sensitive web page. The base rate is often low in realistic WF scenarios. The defender wants to lower the attacker's TPR and precision.

2.5 The One-Page Setting

In previous work, WF defenses were evaluated according to:

(1) Their ability to reduce the recall (TPR) of known attacks, in a multi-page closed-world or open-world setting;
(2) Their ability to reduce the precision of known attacks, in a multi-page open-world setting.

In both cases, the number of positive classes was 100 or higher, and there may be one negative class representing all non-monitored pages. This setting was created to evaluate WF attacks; it is an intentionally difficult setting to allow the attacker to prove his general effectiveness [3, 18]. But the same settings were used to evaluate website fingerprinting defenses [2, 8, 10, 18, 19], without adjusting for the fact that the difficulty of the setting for the attack makes it too easy for the defense to succeed.

In this work, we propose that WF defenses should be evaluated under the **one-page setting**, where there is only one monitored class and one non-monitored class. The monitored class represents one page that the attacker is trained to identify. The attacker's success is measured in recall and precision, and the defender's objective is to lower both. To analyze the defense's effectiveness over a range of pages, we repeat the experiment with different pages as the monitored class, taking the mean performance of the attacker on these pages.

We give two main reasons why the one-page setting is preferable to the multi-page setting for defense evaluation.

First, and most importantly, a defense should be designed to meet a high standard of evaluation. This is a general principle of security and privacy research. A defense that assumes conditions unfavorable to the attacker can only be considered a partial defense.

Second, it is realistic for an attacker to want to monitor only one web page, and that is sufficient to threaten user privacy. A one-page attack can satisfy many use cases:

- Targeted surveillance of a sensitive website to identify certain users or demographics;
- A police force tasked with busting a drug trade network;
- Targeted campaigns (and/or harassment) of users visiting a certain website;
- A popular but embarrassing website that users would not want to be identified as visiting, such as pornographic content;
- A website owner who only wants to track users accessing his own homepage.

Such an attack can have a chilling effect on users' willingness to use anonymity networks and thus erode trust in the technology.

2.6 TPR/FPR Tradeoff

Another aspect that a defense evaluation should include is the tradeoff between TPR and FPR. It is possible to trade off some of the high TPR of known attacks for a lower FPR to achieve better precision by rejecting low-confidence positive classifications [17]. For a defense to claim success, it must take this into consideration as well: it needs to be successful against the entire range of TPR/FPR values an attack can achieve, considering potential tradeoffs.

We will find that for many scenarios, it is not necessary for FPR to be lower than the base rate in the one-page setting, unlike the multi-page setting. In fact, some attacks perform best when the attacker maximizes his TPR (no tradeoff is performed). This is an unexpected consequence of using the one-page setting, and we will explore this in our evaluation.

3 DEFENSE EVALUATION

In the previous section, we argued for the importance of using the one-page setting to analyze WF defense effectiveness. We apply this new methodology to state-of-the-art defenses in this section. In Section 3.2, we determine if these defenses are effective against WF attacks. We find that under the one-page setting, all known defenses, even higher-overhead ones, are unable to lower TPR to a satisfactory degree. We investigate the TPR/FPR tradeoff in Section 3.3, and we are able to reduce the FPR of lower-overhead defenses below 1% while maintaining 90% TPR.

To understand why even the stronger defenses have failed, we will also discuss how previous design paradigms interact with the one-page setting. We separately analyze Decoy (Section 3.4) and Tamaraw (Section 3.5) to determine why they fail in spite of previous work.

3.1 Experimental Setup

We use Gong and Wang's WF data set [8] for our experiments. The data set is relatively recent (2019) and since we are evaluating known attacks and defenses, it is best for us to maintain comparability with previous work. The data set was collected on Tor Browser 8.5a7 on Tor 0.4.0.1-alpha. It contains Alexa's top 100 websites, each visited 100 times, with 10,000 other pages as the non-monitored class. Though it is smaller than some other data sets used to evaluate WF attacks [15], it is sufficient for our purpose as we are performing defense evaluation. This data set was collected with one machine connected to a university network, relying on Tor's random circuit selection for generalizability. Our evaluation of results used a computing cluster (left unnamed for blind review).

3.2 Results

We chose five representative defenses to evaluate: Random, WTF-PAD, Front, Decoy, and Tamaraw. "Random" is a simple benchmark defense that randomly adds dummy packets to the sequence in a uniform fashion. The other defenses were chosen as representatives of different defense paradigms. Sirinam et al. showed success in attacking WTF-PAD with DF [15], and they showed DF to be stronger than competitive deep learning attacks. Front, Decoy, and Tamaraw are not considered "broken" by any attacks.

To test these defenses, we deploy the three WF attacks described in Section 2.2 (kFP, CUMUL, DF) against them in the one-page setting. We show the data overhead (extra data required to load a page) of the defenses to compare their costs; the data overhead is a burden to the network. Tamaraw is the only defense that also delays packets, increasing page load times by 184%.[1] We evaluate both multi-page TPR/FPR and one-page TPR/FPR on Gong and Wang's data set.

In Table 1, we see significantly higher TPR values against all defenses in the one-page setting; the gap is more pronounced when a defense is applied than when there is no defense. kFP performs notably better against Decoy than against Front and Tamaraw in the multi-page setting, but their TPR in the one-page setting is quite similar. Most surprisingly, the one-page setting exposes even Tamaraw to a 91% TPR with kFP, where it only had a 2% TPR in the multi-page setting. Tamaraw was presented as allowing no more than a 10% true discovery rate for most websites [2], and has been frequently shown to be the most robust defense against WF [9, 10, 19].

Among the three attacks, kFP performed the best, and DF did not perform especially well. This is likely due to the small training sets (for DF, only 162 samples for training and 18 for validation) in each classification problem. It may also be because certain hyperparameters in DF are sensitive to the classification problem — in

[1] This number is somewhat higher than prior work, which suggests a page load time increase of around 140% [19], because we are using a more pessimistic simulation that assumes inter-packet times cannot be shorter in the defended trace than in the base trace.

Table 1: Results of three attacks on WF defenses. Multi-page refers to the original multi-page open world methodology, while One-page refers to the one-page methodology we recommend for defense evaluation in this paper.

Defenses	Overhead	Multi-page		One-page					
		kFP		kFP		CUMUL		DF	
		TPR	FPR	TPR	FPR	TPR	FPR	TPR	FPR
None	0%	91.3%	3.4%	99.1%	0.9%	97.7%	4.9%	86.5%	5.5%
Random	22%	49.8%	5.7%	97.6%	4.7%	96.5%	7.5%	83.9%	8.8%
WTF-PAD [10]	32%	60.3%	14.8%	97.6%	4.4%	5.5%[1]	0.3%	66.6%	73.6%
Front [8]	67%	18.5%	7.5%	92.9%	13.1%	80.5%	22.6%	76.2%	27.3%
Decoy [12]	98%	30.8%	9.2%	91.2%	10.0%	77.5%	26.7%	73.0%	39.6%
Tamaraw [2]	107%	2.9%	4.4%	91.0%	21.4%	91.1%	21.6%	59.8%	38.9%

[1] The result for CUMUL on WTF-PAD is not in error; it is due to a failure of the SVM to converge based on preset parameters. While other parameters may produce better results, we kept this result as it showed a limitation of SVMs and did not particularly affect any of our other results, which would be derived with kFP.

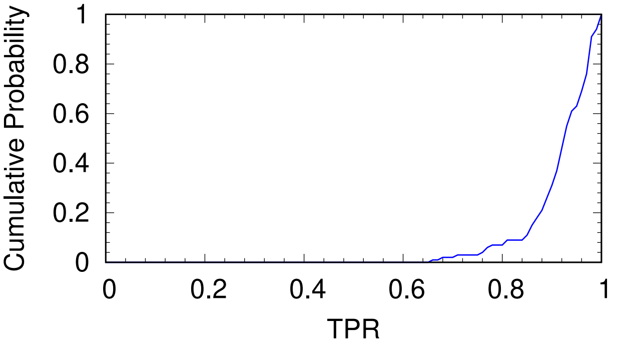


Figure 1: CDF for TPR of kFP against Tamaraw for 100 different web pages.

fact, we had already lowered its training batch size parameter from 128 to 5, or the classifier could not be trained. While it is possible that further optimizations to DF could improve results, our intent is to evaluate defenses (not attacks), and the strong performance of kFP is sufficient to do so.

Due to kFP's superior performance compared to CUMUL and DF for our scenario, from this point on, we evaluate defenses using only **kFP** as a benchmark.

We want to determine the variance in performance against different pages in the one-page setting. We measure the TPR of kFP against Tamaraw (the hardest defense) for each of 100 pages individually and plot the results as a CDF in Figure 1. The standard deviation of TPR across the 100 pages is fairly low at 6.9%, and Figure 1 displays this phenomenon: no page had a TPR lower than 66%, and only 9 pages had a TPR lower than 85%. Broadly speaking, no page in our data set is particularly safe in the one-page setting even when protected by Tamaraw.

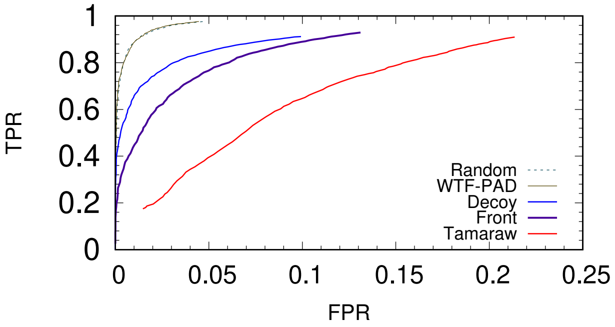


Figure 2: TPR and FPR for kFP when traded off based on classifier confidence. Note the FPR scales up to 0.25. When classifier confidence is ignored, the result would be the furthermost top-right point on each curve. The results for Random and WTF-PAD overlap each other.

3.3 TPR/FPR Tradeoff

Decreasing the FPR increases precision, and it has been argued that high precision is important for website fingerprinting [17]. The FPR values we found in Table 1 are relatively high for Front, Decoy and Tamaraw. We investigate if the FPR can be reduced by trading off a portion of their high TPR values. For a defense to claim success, it needs to show that it succeeds against the entire range of the attacker's possible TPR/FPR values.

For the TPR/FPR values we obtained above, classifiers treated the two classes (monitored and non-monitored) equally. We can reduce FPR with the simple but effective technique of increasing the minimum threshold confidence (or class probability) required to classify a trace as monitored, depending on the classifier. We apply this technique and show the results in Figure 2.

The results show us that it is indeed possible to significantly reduce the FPR incurred against each defense. We can lower the FPR close to 0% when there is no defense (as was shown in previous work [15, 17]), and it is even possible to do so with defenses. The highest TPR at which less than 0.1% FPR could be measured was 63% for WTF-PAD, 42% for Decoy, and 20% for Front.

At the highest confidence settings, Tamaraw holds out as the strongest (though most costly) defense against kFP, at 17.5% TPR and 1.5% FPR; poor results for the attacker. The attacker achieves 37% TPR in the one-page setting compared to 2.9% TPR in the multi-page setting, if we hold the FPR at 4.4% in both cases. The TPR/FPR tradeoff is not especially effective for Tamaraw, as shown by the relatively straight line in Figure 2 (a straight line with slope 1 would indicate the trivial tradeoff).

3.4 Decoys do not Force a 50-50 Guess

The use of the one-page setting has the added benefit of exposing implicit assumptions that were not previously examined. Decoy is a simple defense that loads a fake decoy page whenever a real page is loaded. Since the WF attacker can at best identify both pages being loaded together, and there is no way to know which of the two is real (as both pages are in fact loaded), it is tempting to conclude that the attacker is forced into a 50-50 guess and can therefore achieve no more than 50% TPR under Decoy. However, this is contradicted by our 91% TPR against Decoy.

The reason for this contradiction is that the set of decoy pages cannot be assumed to come from the same distribution as real pages being visited by the client. This is because different clients need to use decoy pages from the same distribution (or their choice of decoy pages alone could identify them), but they still visit real pages differently. We replicate this effect in the experimental setting by setting aside a portion of pages as a decoy page set, and loading both monitored and non-monitored pages with randomly chosen decoy pages from this set. As a result, packet sequences of monitored pages still look more similar to each other than they do to packet sequences of non-monitored pages.

To reduce the accuracy of the attacker to no more than 50%, the client would have to always use the monitored page as the decoy when visiting non-monitored pages, which is impossible as the attacker decides which page to monitor.

From another perspective, we observe that if an attacker is trained to monitor a specific sensitive page, and the attacker sees that the client has visited two pages, among which one is the sensitive page, the attacker's ideal strategy is *not* to guess that the sensitive page was visited 50% of the time — he should classify it as sensitive much more often than that. This strategy works because the chance that a sensitive page would be used as a decoy page is usually much smaller than the chance that a sensitive client would visit a sensitive page; all clients do not visit the same page at the same base rate. The analysis needs to consider the fact that sensitive accesses are not uniformly distributed among all clients.

3.5 Tamaraw and Anonymity Sets

Tamaraw *regularizes* the packet sequence, fixing packet rates, so that the resulting packet sequence is defined by only one feature — the sequence length. All packet sequences of the same length will be identical to each other, so they can be considered to be in the same anonymity set, and larger anonymity sets are created by padding the sequence length to multiples of a fixed integer. In the multi-page setting, an attacker cannot distinguish within the large and diverse anonymity sets created by Tamaraw. But in the one-page setting, Tamaraw failed, even though we tested a strengthened version of Tamaraw that pads sequence lengths to multiples of 500 (instead of 100 in the original work [2]). We investigate why by exploring its anonymity sets.

In Figure 3, we show a scatter plot of anonymity set sizes and how many positive elements each contained. For classification, we use a simple strategy of identifying anonymity sets and classifying each anonymity set to the majority of elements it contained. This strategy is Pareto-optimal for the attacker and would achieve a TPR of 0.925 with an FPR of 0.176 (similar to kFP and CUMUL).

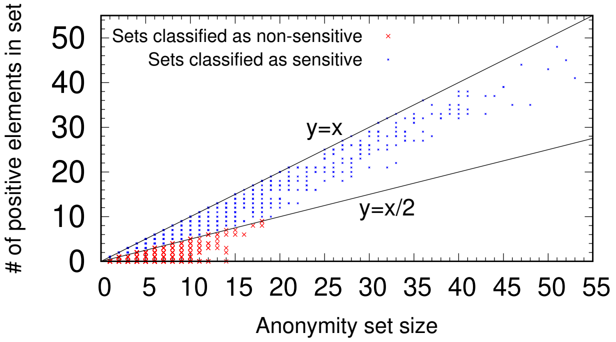


Figure 3: Scatter plot of anonymity set sizes and the number of positive elements in them, as well as their classification using a simple majority strategy delineated by $y = x/2$.

Here, we see that sensitive and non-sensitive pages belonged to anonymity sets of very different sizes. The maximum size for a set classified as non-sensitive was 19, while that of sensitive sets was 53 (out of 200 elements in each classification problem). 79% of sensitive pages belonged to anonymity sets of size 9 or above, while only 26% of non-sensitive pages did so. Overall, 1998 out of 10000 sensitive pages belonged to anonymity sets that only contained sensitive pages, while 6602 out of 10000 non-sensitive pages (a majority) belonged to anonymity sets that only contained non-sensitive pages. These are not truly anonymity sets, and they do not confuse the attacker.

Non-sensitive sets were smaller because sequences of a sensitive page were more similar to each other than non-sensitive pages were to each other. Across our data set, the mean coefficient of variation for the sequence lengths of a sensitive page was 0.196, compared to 0.649 for non-sensitive pages. As a result of their similarity, sequences of a sensitive page would be grouped together in the same anonymity sets. To strengthen Tamaraw in the one-page setting, either greater anonymity sets or more randomness is required; we explore these options later when attempting to derive a stronger defense in Section 5.1.

4 WEBSITE FINGERPRINTING SCENARIOS

The standard laboratory scenario for WF attacks is a basic supervised classification problem: the attacker is presented with labelled testing elements, and his performance is evaluated by his overall TPR and FPR. This standard scenario does not fully capture a real attacker's objective in WF, and it is not obvious how such an attacker's TPR/FPR would translate to a realistic threat. To provide a more complete WF analysis, we define and investigate three WF scenarios in this work. These scenarios allow us to determine if the TPR/FPR values we found in the previous section would allow an attack to succeed against the defenses. They will also allow us to re-examine the implicit assumptions of the standard laboratory scenario.

We will explore the following three scenarios:

- The **selection scenario** (Section 4.1), where the attacker, monitoring many clients, picks out which ones are visiting a sensitive page.

- The **identification scenario** (Section 4.2), where the attacker, monitoring a single client, decides if she is visiting a sensitive page or not.
- The **linking scenario** (Section 4.3), where the attacker observes a visit to a sensitive page, and tries to determine which of several clients did so.

4.1 Selection scenario

In the selection scenario, the eavesdropper monitors a large number of clients, and a portion of them are visiting sensitive pages. He wants to pick these clients out of the larger group as candidates for further action, such as to identify members of a compromising website or specific interest groups. Results of WF in the selection scenario can allow him to decide where to apply more powerful but resource-limited surveillance techniques, such as zero-day malware, phone tapping, or electromagnetic monitoring.

We define the scenario as follows. S clients visit K pages each, among which N clients ("sensitive clients") have visited a sensitive page M times each. The other $S - N$ clients do not visit the sensitive page. The eavesdropper wants to select N' clients that are visiting sensitive pages. To do so, he uses WF to classify all page visits, and selects the N' clients that have visited the most pages.

Analysis

We examine a setting with $S = 1000$ clients among which $N = 30$ clients visit sensitive pages. The attacker observes $K = 2000$ page visits for each client in total, and the sensitive clients visits $M = 60$ sensitive pages. The attacker attempts to guess who they are by selecting the top $N' = N = 30$ clients by sensitive page access count, where the count is determined by the classifier. Setting the number of actual sensitive clients to be the same as the attacker's number of selections allows us to simplify the analysis by using a single accuracy value to measure success, and it gives the attacker the hardest possible task without making it inherently impossible. (The attacker does not need to know N.) As 3% of clients are visiting a sensitive page 3% of the time, overall, clients are only visiting the sensitive page at a very low base rate of 0.09% — below the lowest base rate examined in previous work [17]. We set a low base rate so that the scenario is difficult for the attacker, in order to show that defenses still do not succeed in this scenario.

We show the attacker's success rate, defined as the percentage of sensitive clients correctly identified as such, against four defenses in Figure 4. The lines show how a TPR/FPR trade-off by increasing the confidence limit would increase the success rate. Indeed, each attack sees an increase in success rate due to the trade off: WTF-PAD (100% → 100%), Decoy (76% → 100%), Front (84% → 100%), but not so much for Tamaraw (53% → 59%). It is also disadvantageous to increase the confidence limit too much, which would cause the success rate to drop due to low accuracy.

The TPR/FPR trade-off is useful for this scenario because the FPR is a stronger determinant of attacker success than TPR (in terms of absolute value). An attacker achieving 90% TPR and 5% FPR has the same success rate of 97.7% as an attacker achieving 80% TPR and 3.9% FPR. Interestingly, we found that the Tamaraw attacker does not benefit much from a TPR/FPR tradeoff, even though his FPR against Tamaraw was 20% (much higher than the base rate).

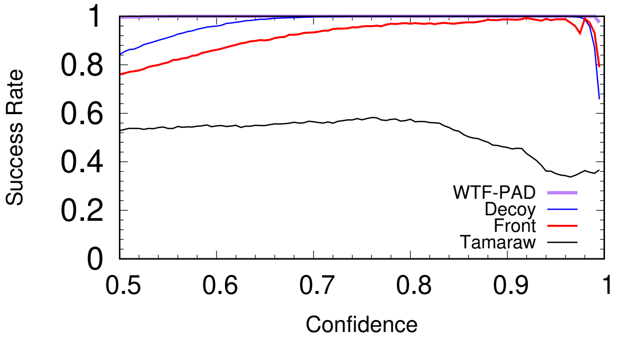


Figure 4: Selection scenario: Success rate (percentage of sensitive clients correctly identified) using kFP against four defenses for $M = 30$, $N = 30$, $S = 1000$ and $K = 2000$, varying the confidence limit for a TPR/FPR tradeoff. 0.5 confidence is equivalent to no tradeoff (maximum TPR).

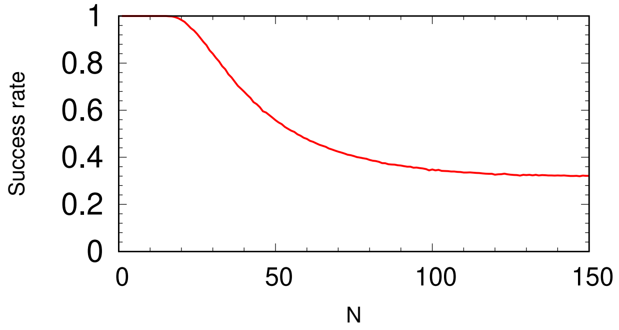


Figure 5: Selection scenario: Success rate against Decoy (91% TPR, 10% FPR), while varying N (the number of sensitive clients), and keeping M (the number of sensitive pages each sensitive client visits) at $M = 1800/N$ to hold the overall base rate constant at 0.09%. The attacker makes $K = 2000$ observations of page visits on $S = 1000$ total clients.

Without a TPR/FPR trade-off, the attacker is still decently successful while incurring a much higher FPR against these defenses than the base rate.

Concentration of base rate

Specifically, it is noteworthy that the eavesdropper can achieve a high success rate with a relatively high FPR compared to the base rate; a 5% FPR would be more than 50 times the base rate. This may seem contrary to the standard laboratory scenario in which a classifier fails when its FPR far exceeds the base rate: this is because, in our scenario, the classifier's goal is not to identify all page visits, but rather to distinguish between a small group with 3% base rate and a large group with 0% base rate. (Here we refer to the former as the *specific base rate*.) Members of the small group are selected based on their perceived number of sensitive actions by the attacker, and so both a high TPR and a low FPR are important.

We examine the effect of the specific base rate in the following. Fixing the TPR at 91% and FPR at 10% (similar to Decoy), we compute the attacker's performance in selecting N clients visiting $1800/N$

sensitive pages, varying N, out of a group of 1000 clients.[2] This fixes the overall base rate at 0.09% while varying the specific base rate of the group to be selected. We show the results in Figure 5, where we see that the specific base rate has a powerful effect on the success rate of the attacker, ranging from above 98% below $N = 20$ (4.5% specific base rate) down to 35% at $N = 100$ (0.45% specific base rate). A larger specific base rate means fewer sensitive clients, but they visit the sensitive page more frequently. There is a large slide in success rate between $N = 20$ and $N = 50$. (Note that as N increases to 150, the random chance of guessing a sensitive client correctly increases linearly to 15%, so the success rate with high N is partly due to random guesses.)

This adds a previously unaddressed factor to the discussion of how the base rate affects the attacker — whether sensitive accesses are concentrated in a few clients or spread among many. Even if the two cases have the same overall base rate, it is far easier for the attacker to detect sensitive clients in the former case. One implication is that clients who only visit sensitive pages with Tor place themselves at greater risk of being selected with WF; clients who use Tor for both sensitive and non-sensitive activities are not as easily detected.

We note that this scenario does not necessarily assume that there are only two groups, one with a high base rate and one with a zero base rate. This is because the attacker's success rate on each group is independent of the existence of other groups. When there are multiple groups with different base rates, the attacker's success rate can be separately derived on each group.

If we had increased N without decreasing M (causing an overall increase in base rate), the attacker's success rate would increase slightly, from 84% at $N = 30$ to 90% at $N = 100$. While the attacker would need to identify more clients, the overall task is easier as more clients are sensitive. In fact, the increase in success rate is almost identical to the increase in the random guessing success rate (from 3% at $N = 30$ to 10% at $N = 100$). This shows that the value of N by itself does not explain the above result; it is indeed due to the concentration of base rate.

The benefit of more observations

Another interesting factor determining the attacker's success in this scenario is the total number of observations the attacker makes. If the attacker can monitor the client for a longer period of time (collecting more page accesses), he will naturally be able to classify the client more accurately.

We examine this effect in Figure 6, where we scale up K and M proportionally while keeping the specific base rate of sensitive clients the same at 3%, based on our attack's TPR/FPR against Tamaraw, the strongest defense. We see the effect of K on the success rate is drastic: from a success rate of 34% when $K = 1000$, the attacker's success rate increases beyond 98% above $K = 10000$. On the other hand, if the attacker could only observe $K = 100$ accesses (i.e. only $M = 3$ sensitive accesses), the success rate is only 7%. The long-lived guard policy of Tor implies that Tor guards can practically collect large numbers of traces on clients connecting to them, as they would have observation periods lasting months.[3]

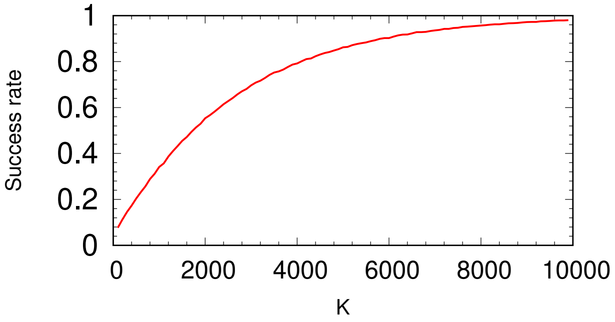


Figure 6: Selection scenario: Success rate against Tamaraw (90% TPR, 20% FPR) while varying K, the number of page accesses observed, and keeping $M = 0.03K$ to hold the base rate constant.

4.2 Identification scenario

We flip around the selection scenario to consider an attacker that wants to make a single yes/no decision on whether or not a specific client has been visiting sensitive pages. In the identification scenario, the eavesdropper has been monitoring the client for a long time, and has collected some traces each corresponding to a single page access. The eavesdropper wants to know if the client has visited certain sensitive sites, such as sites of a specific political ideology, whistleblowing sites, or online marketplaces. Identifying the client this way may give the eavesdropper sensitive targeting information for purposes of surveillance, recruitment, harassment or ostracization. Out of K pages, the client has either visited M monitored pages, or has visited 0 monitored pages. The eavesdropper faces a yes/no decision problem on whether or not the client has visited monitored pages. The identification scenario was described in earlier work [17] and we perform a more complete analysis here.

Analysis

We represent the number of detected monitored pages with the variable x_{mon}. If the client does not visit monitored pages, $Pr(x_{mon} > L)$ is given by the binomial distribution of K trials with success rate FPR; it is one minus the CDF up to $x = L$. For a sensitive client who visits M monitored pages, $Pr(x'_{mon} > L)$ is given by the sum of two binomial distributions, where there are M trials of success rate TPR and $K - M$ trials of success rate FPR. The eavesdropper decides that the client is one to visit monitored sites if he observes more than L visits. Therefore, $Pr(x_{mon} > L)$ is the attacker's false positive rate (FPR_{id}) and $Pr(x'_{mon} > L)$ is the attacker's true positive rate (TPR_{id}) in the identification scenario.

Based on an attack with 90% TPR and 20% FPR (similar to Tamaraw) that can observe 1000 page accesses for the client, we chart a range of TPR_{id} and FPR_{id} values by varying L in Figure 7. We see that this attack would perform poorly against a $M = 10$ (1% base

[2] If $1800/N$ is not an integer, the client randomly chooses between $\lfloor 1800/N \rfloor$ and $\lceil 1800/N \rceil$ so that it averages to $1800/N$.

[3] If Tor guards were short-lived, it would take them longer to collect as many traces on their clients depending on their share of the global guard bandwidth; however, they would be able to do so for a greater number of clients over time, so we are not necessarily recommending a short-lived guard policy.

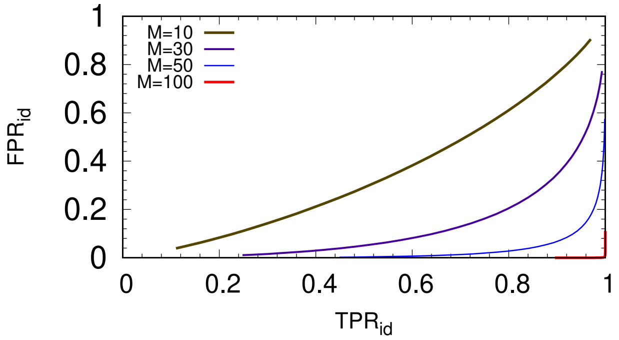


Figure 7: Identification scenario: TPR and FPR in the identification scenario (TPR_{id}, FPR_{id}) based on a WF attack with 90% TPR and 20% FPR (similar to kFP against Tamaraw), given 1000 observations out of which a sensitive client visits the sensitive page M times. The line for $M = 100$ is nearly covered by the x-axis as FPR_{id} is close to 0.

rate) client, but it would perform well against a $M = 50$ (5% base rate) client, even though the FPR of the underlying attack is 20%. At $M = 50$, the attacker can identify 67% of sensitive clients while mis-identifying 1% of non-sensitive clients. At $M = 100$, TPR_{id} rises to 97% while FPR_{id} drops to 0.1%.

Just as we had observed for the selection scenario, the distribution of base rate among clients would also affect our results. We found that in the above setting, if 10% of clients visited the sensitive page 1% of the time, we could identify 51% of sensitive clients while mis-identifying 30% of non-sensitive clients as sensitive; a poor result. But if 1% of clients visited the page 10% of the time, we could identify 96% of these clients while mis-identifying less than 0.01% of non-sensitive clients.

While the results are dependent on M and L, an attacker does not need to know M to set L; he can do so based on his knowledge of his classifier's FPR by setting L to be comfortably higher than the FPR times K.

How many sensitive pages can you visit before detection?

A client can make a small number of sensitive page visits without detection, as due to the false positive rate, she may be statistically indistinguishable from someone who has visited no sensitive pages. We want to find how many sensitive pages (M) a client can make out of a total of K page visits without being detected by the attacker. Here, we say that the attacker successfully detects the client if he achieves above 90% TPR_{id} and below 1% FPR_{id}. We consider three defenses, Front, Decoy and Tamaraw in four settings: $K = 20$, $K = 50$, $K = 100$, $K = 1000$, and show the results in Table 2.

When the number of pages visited is as small as $K = 20$, around half of the page visits need to be sensitive for the attacker to confidently classify the client as a sensitive one. As K increases, the number of sensitive pages the client can visit does not increase proportionally; at $K = 1000$, only a 4% to 7% base rate is required for the attacker to detect the client.

It is worth noting that the number of sensitive pages a client can safely visit is only about 50% higher under Tamaraw than Front, although attack performance under Tamaraw is much worse. The similarity of these results is largely due to their similarity in TPR (all within 90–93%). We investigate this further by performing a TPR/FPR tradeoff on Front (TPR: 0.912 → 0.801, FPR: 0.100 → 0.069) for a 30% decrease in FPR, labelling the result under Front* in Table 2. This slightly *increases* the number of pages a client can visit in low K settings, and decreases it in high K settings; the overall change is very small despite the large change in FPR. The results suggest that a high TPR/FPR attack can perform well in the identification scenario, even with a low base rate.

Table 2: Identification scenario: How many sensitive pages a client can visit before an attacker has a high likelihood ($TPR_{id} > 0.9$, $FPR_{id} < 0.01$) of detecting them as such, depending on how many total pages a client will visit (K). Results are based on TPR/FPR values achieved by kFP against three defenses, Front, Decoy, and Tamaraw. The Front* row is based on results against Front after a TPR/FPR tradeoff.

	$K = 20$	$K = 50$	$K = 100$	$K = 1000$
Front	8	10	16	44
Front*	9	11	14	41
Decoy	9	13	16	49
Tamaraw	11	16	22	69

4.3 Linking scenario

In the linking scenario, the eavesdropper knows that a sensitive page access is one of only several potential candidates. This may be because he knows that the sensitive page visit was made at a specific time through an observation outside of the anonymity network; having broad tapping capabilities over the anonymity network, he narrows it down to one out of several traces that happened at that time. For example, this page visit may be whistleblowing/leaking activity, a social media post, or a threat sent by e-mail. The eavesdropper wants to link the observed page visit with the specific trace, which in turn tells him who was the source of that activity.

It is worth noting two interesting aspects of this scenario. First, the base rate is not relevant insofar as it does not change the total number of candidate traces. Unlike the previous scenarios, a client cannot protect herself by lowering her base rate; she can only be protected by other users who are also visiting web pages at the same time (and possibly location). Second, the one-page setting is a natural fit for the linking scenario, as there is no motivation for the attacker to link any other pages.

Let the number of potential candidate traces be P, among which one is the true sensitive access. If one or more traces are classified as a sensitive page visit, the attacker links the highest-confidence classification with the visit. If all traces are classified as non-sensitive, the attacker links the lowest-confidence non-sensitive classification instead. The success of the attacker is dependent on the quality of the classifier's confidence ratings.

Analysis

We set up the experiment by randomly choosing one sensitive trace and $P - 1$ non-sensitive traces out of our data set. We measure the attacker's success rate as the chance that the attacker correctly identifies the sensitive trace. We repeat this experiment 10,000

Table 3: Linking scenario: The chance that the attacker identifies which one out of P traces is sensitive, using kFP's confidence metric to obtain the trace most likely to be classified as sensitive.

	$P = 5$	$P = 10$	$P = 20$	$P = 100$	$P = 500$
None	1.00	1.00	0.99	0.98	0.96
Random	0.98	0.96	0.93	0.81	0.6
WTF-PAD	0.98	0.96	0.93	0.82	0.63
Front	0.87	0.78	0.66	0.41	0.24
Decoy	0.91	0.85	0.78	0.59	0.44
Tamaraw	0.70	0.54	0.37	0.17	0.14

times for each defense and each value of P. In Table 3, we show the overall success rate of kFP against various defenses for $P = 5$, $P = 10$, $P = 20$, $P = 50$, $P = 100$, and $P = 500$.

When there is no defense, the attacker can link the sensitive trace with the sensitive visit at very high probability even when there are 500 candidate traces. Given 500 traces, the attacker can also succeed most of the time against Random and WTF-PAD and almost half of the time with Decoy. The result with Decoy affirms once more that Decoy-disguised traces still retain much of the true trace's features. While Front is not as effective as Decoy in terms of TPR/FPR, it is more effective at disguising the true trace in the linking scenario.

For the highly costly Tamaraw, the attacker can still succeed most of the time up to $P = 10$. On the other hand, the attacker is not likely to identify the one correct trace out of $P = 500$ traces. In fact, the attacker is only 20% likely to find the correct trace within his top 10 guesses; the same probability is 74% for Decoy. Nevertheless, the $P = 500$ result is still 70 times better than random guessing.

To validate the usefulness of confidence in the linking scenario, we consider what the attacker's accuracy would be if he simply attempted to classify all traces using kFP, and if there were multiple positive classifications (or all classifications were negative), guessed one of them randomly. Setting $P = 20$, the attacker's accuracy would drop to 0.91 (compared to 0.99) with no defense; 0.65 (compared to 0.93) with WTF-PAD; and 0.34 (compared to 0.66) with Front. This result shows that when evaluating defenses in the linking scenario, a confidence metric helps demonstrate the attacker's true capability. The attacker's strong results in the linking scenario can be said to be a consequence of the distinctive confidence between positive and negative classifications.

5 STRENGTHENING THE DEFENSES

Our experiments under the one-page setting show that known website fingerprinting defenses cannot prevent the attacker from classifying a web page, whether in the standard laboratory scenario or in several scenarios designed for realistic attacker goals. We want to see if known defenses could be fortified along their original design to meet this higher standard of evaluation; if not, new defenses would have to be created. We investigate parametric adjustments and changes to inject randomization into three defenses, Tamaraw, WTF-PAD, and Front, and re-evaluate their performance in the one-page setting.

5.1 Tamaraw

Tamaraw enforces (different) fixed packet rates on both parties and pads the end of communication to a multiple of a parameter ℓ. After regularizing the packet rate, if the smallest multiple of ℓ greater than the total number of cells (including dummy cells) is $A\ell$, sequence-end padding will pad it to $(A + k)\ell$ drawing k from the geometric distribution $Pr(X = k) = (1 - p)^k p$. For convenience we write $\ell = 500L$ and $p = 1/(G + 1)$ so that L and G are small integer parameters for Tamaraw. In previous experiments, $L = 1$ and $G = 1$.

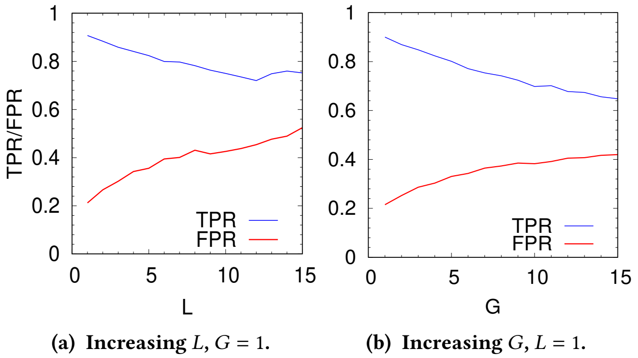
Increasing these two parameters gives us two different paradigms for how to fortify a defense in the one-page setting. Increasing L decreases variation between different sequences, making it more likely that two different pages will produce the same sequence. Increasing G, on the other hand, increases random variation, so that different sequences of the same page are more likely to produce different results. Both are potentially able to confuse the attacker at a greater cost to data overhead. (Note that sequence-end padding does not increase page loading time as the client is not forced to wait for sequence-end padding to finish before loading a second page.)

We show the distinct effects of increasing L and G on the performance of kFP under the one-page setting in Figure 8a and Figure 8b respectively. When increasing L, G is fixed at 1; when increasing G, L is fixed at 1. We observe that the TPR decreases and FPR increases in both cases, narrowing the gap between the two and rendering the classifier ineffective. In other words, it is possible to strengthen Tamaraw in the one-page setting whether by increasing L or G. One oddity is a slightly *increased* TPR at high values of L, though the classifier does not perform better overall due to the corresponding increase in FPR.

To enable a direct comparison between these two strategies, we plot the gap between TPR and FPR against the data overhead of increasing L and G in Figure 9. We see that increasing G is a more efficient way to strengthen Tamaraw in the one-page setting, and the difference is especially pronounced at higher overheads. At 200% overhead through increasing L, we can reduce attacker effectiveness to 80% TPR and 40% FPR; through increasing G, we can further reduce it to 70% TPR and 38% FPR. This suggests that, for Tamaraw, increasing randomness is more cost-effective against the one-page attacker than fixed deterministic padding, though the optimal method may involve some combination of L and G.

In Section 3.5, we showed that Tamaraw failed under the one-page setting because the sequence lengths of a monitored page tended to occupy its own anonymity sets while non-monitored pages were scattered. We re-examine the anonymity sets in $L = 9, G = 1$ and $L = 1, G = 9$ in Figure 10, compared with original Tamaraw. Here we show the anonymity sets of site 1 compared to 100 non-monitored pages.

Figure 10 shows that in original Tamaraw, it is clear that the attacker can achieve success by classifying sequences between 3500 to 6000 cells as positive, and all others as negative. Increasing L to 9, there are only 8 anonymity sets left, and they are mostly evenly divided between positive and negative cases. On the other hand, increasing G to 9, we observe the sequence lengths of both sets



(a) **Increasing** L, $G = 1$.

(b) **Increasing** G, $L = 1$.

Figure 8: TPR/FPR of kFP against Tamaraw with (a) increasing L (greater deterministic end-of-sequence padding), and (b) increasing G (more random end-of-sequence padding).

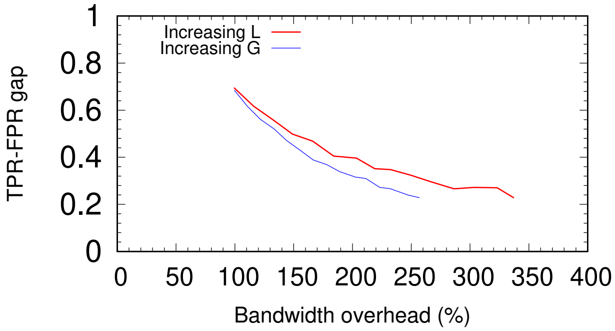
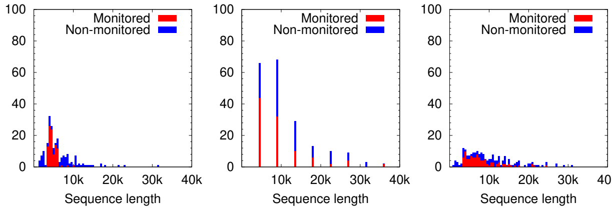


Figure 9: kFP against Tamaraw measuring the TPR minus FPR gap against data overhead as a percentage. Here, 100% data overhead means doubling the expected data of browsing; this is incurred by original Tamaraw.



(a) $L = 1$, $G = 1$

(b) $L = 9$, $G = 1$

(c) $L = 1$, $G = 9$

Figure 10: Histogram of Tamaraw anonymity sets based on total number of Tor cells for 100 instances of site 1 and 100 non-monitored instances. (a) is original Tamaraw.

being dispersed across possible values. These two distinct strategies are both able to confuse the attacker.

We study whether or not these improvements would defeat a one-page attacker in the selection scenario. Like before, the attacker attempts to select $N = 30$ clients from $S = 1000$ where sensitive clients visit the page at a base rate of 3%. We examine two cases of the number of observed packet traces, $K = 2500$ and $K = 10000$. The results are in Figure 11; they confirm that the attacker can indeed be defeated under the selection scenario, even with a large number of observations ($K = 10000$). The difference is especially stark under $K = 10000$, dropping from 97% success rate at 100% overhead (original Tamaraw) to 26% at 250% overhead when increasing G. In both cases, increasing G is more efficient.

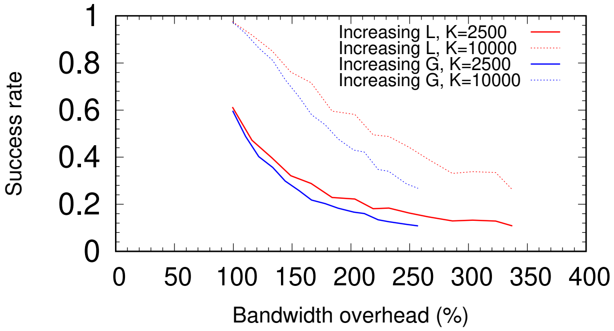


Figure 11: kFP against Tamaraw measuring success rate under the selection scenario for $K = 2500$ and $K = 10000$, varying M to keep base rate at 3%. For all lines, $S = 1000$, $N = 30$.

Our results shows that while it is possible to defeat the attacker in the one-page setting using modified Tamaraw, cheaper options still need to be explored. The overhead values we obtain are not practical for general deployment to all Tor users.

5.2 WTF-PAD

WTF-PAD [10] is based on Adaptive Padding, which focuses on eliminating inter-packet timing as a feature by inserting dummy packets. It does so by mimicking expected inter-packet times (IPTs) from a target distribution, which could be learned from real traces. It was shown to cost little overhead and was effective against several WF attacks, but later WF attacks based on deep learning defeated it [15]. In this work, we also showed that it was not effective in the one-page setting. Similar to before, we investigate if strengthening the defense by increasing its overhead would allow it to succeed.

WTF-PAD has no explicit parameters except those that describe its target IPT distribution. Shorter IPTs increase the overhead as more dummy packets are generated. To increase its overhead, we alter the sampling process from this distribution. [4] We divide all sampled IPTs by a fixed number D, maintaining the original randomness of WTF-PAD but directly increasing the overhead. We test five settings for D, and show the results in Figure 12 plotting the TPR/FPR of kFP against the overhead. The results show that our fix for WTF-PAD is largely unsuccessful at defeating kFP in the one-page setting. Even with 293% overhead, the TPR only drops from 0.973 to 0.960, and the FPR increases from 0.045 to 0.062.

These results show that not all methods of adding dummy packets are equal; not all defenses can be strengthened for the one-page setting. This may be because any method of adding overhead to WTF-PAD compromises its original design principle of mimicking real IPT distributions.

[4] We use the default `normal_recv` distribution in this and previous experiments.

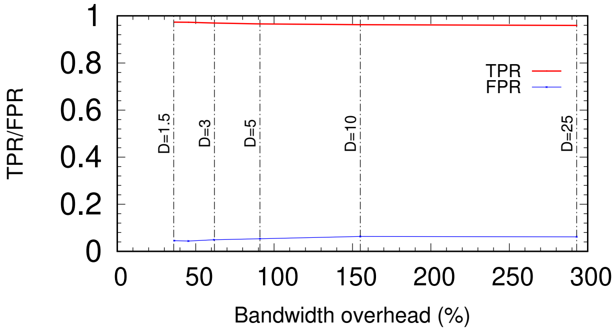


Figure 12: kFP against WTF-PAD when increasing the IPT divisor, D.

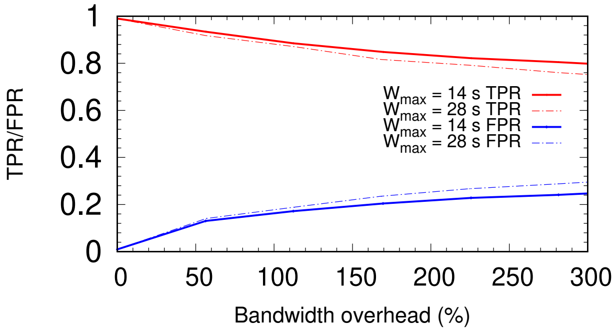


Figure 13: kFP against Front when increasing the maximum for the number of random packets, N_{max}, under two settings, $W_{max} = 14$ s and $W_{max} = 28$ s.

5.3 Front

While Tamaraw shows more promise for the one-page setting than WTF-PAD, it delays user packets (unlike WTF-PAD), thus degrading browsing performance. We turn to Front as it is also a zero-delay defense like WTF-PAD, and it was able to thwart several attacks that WTF-PAD could not [8].

Front focuses on adding random dummy packets to the front of the sequence based on two values: N, the maximum number of dummy packets to add, and W, a parameter that controls where most of the packets will be added according to a Rayleigh distribution. For each sequence, N is randomly picked between 1 and N_{max}, and W is randomly picked between 0 s and W_{max}. We choose two settings for $W_{max} = 14$ s and $W_{max} = 28$ s, and six settings of N_{max} from $\{2500, 5000, 7500, 10000, 12500, 15000\}$.

We show the results in Figure 13. The results show some promise in decreasing the performance of kFP in the one-page setting. At 200% overhead, we reduce the attacker to 80% TPR and 25% FPR, compared to 70% TPR and 38% FPR for our improved Tamaraw. The $W_{max} = 28$ s line is distinctly better at defending against kFP; a higher W spreads the packets out more evenly and thus more randomly.

These results suggest that the high standard set by the one-page setting can also be met by zero-delay defenses, but the required data overhead may be very large. So long as the network can tolerate the extra data, the impact on user performance will be minimal. On the other hand, the advantage of using Tamaraw is that an analysis of anonymity sets can give certain guarantees on the upper bound performance of *any* WF attack, which we cannot obtain with current zero-delay defenses.

6 DISCUSSION

6.1 Varying the number of pages

In this work, we propose to evaluate defenses under the one-page setting instead of the multi-page setting, which is more suitable for attack evaluation. A middle ground between the two settings is the binary multi-page setting: the attacker wants to monitor access to a number of pages, but does not care which particular page is being accessed. For an attacker who wants to build up a profile for the user's interests and beliefs, this is more powerful than the one-page setting.

In Table 4, we vary the number of monitored pages in the binary multi-page setting and measure the kFP attack's TPR and FPR against the same defenses we tested before. The attacker only needs to determine if the page is monitored or not. We see that for most defenses, an increase of the number of monitored pages up to 50 only slightly decreases TPR. We especially note that the difference between 20 monitored pages and 50 monitored pages is small. Only Tamaraw regains much of its defensive capability, as the spread of monitored pages forces the attacker to frequently make false positive errors. We can therefore conclude that for most defenses, even if the attacker were to monitor access to 50 pages in the binary setting, they could still succeed at high probability.

How many pages the attacker would monitor depends on the attacker's needs. One interesting caveat is that because Tor does not save browsing history and does not cache cookies, many users visiting a website would have to go through its front page instead of jumping to a stored page or logging in automatically. Monitoring only the front page can therefore be a useful way to capture accesses to an entire website on Tor. Regardless of these results, a WF defense should be designed to be strong enough to defeat an attacker monitoring one page.

6.2 What Makes the One-Page Setting Difficult

Why did all evaluated defenses fail in the one-page setting? Broadly, there can be two sources of difficulty for the one-page setting:

(1) When there is only one class, the classifier can learn to be bolder in classifying for that class;
(2) A reduction in the total number of positive classes by itself increases TPR and reduces FPR.

We can analyze each of these effects by performing an extra experiment where the multi-page classifier is used to classify in the one-page setting.[5] Comparing the one-page and multi-page classifiers' performance on the one-page setting will reveal the first effect, and comparing the multi-page classifier's performance on the one-page and multi-page settings will show the second effect.

We focus on Front for this experiment. We perform the additional experiment on Front and show the results in Table 5. To enable a comparison between the two classifiers in the one-page setting, we set a higher confidence limit for the one-page classifier to obtain the same FPR as the multi-page classifier. The one-page classifier

[5] The converse, using the one-page classifier in the multi-page setting, is not valid.

Table 4: TPR/FPR while varying the number of monitored pages in the one-page setting.

Defenses	Number of monitored pages									
	2		5		10		20		50	
	TPR	FPR	TPR	FPR	TPR	FPR	TPR	FPR	TPR	FPR
None	99.4%	1.3%	98.6%	1.4%	98.5%	1.6%	98.6%	1.5%	98.2%	1.8%
Random	97.9%	6.2%	95.5%	8.1%	94.3%	9.2%	93.1%	10.1%	91.6%	10.6%
WTF-PAD [10]	96.2%	4.9%	96.2%	6.5%	95.3%	7.0%	94.6%	7.7%	93.5%	7.7%
Front [8]	91.3%	15.1%	84.7%	18.3%	83.1%	17.5%	81.1%	18.8%	79.8%	17.3%
Decoy [12]	89.8%	11.5%	88.7%	12.6%	87.8%	13.1%	87.2%	13.6%	86.0%	13.2%
Tamaraw [2]	85.1%	28.8%	76.0%	36.8%	68.7%	37.7%	60.3%	36.2%	51.4%	31.3%

Table 5: TPR/FPR of the one-page/multi-page classifier in the one-page/multi-page setting.

Classifier	Setting	TPR	FPR
One-page	One-page	.929	.131
One-page	One-page	.320	.003
Multi-page	One-page	.220	.003
Multi-page	Multi-page	.185	.075

then achieves a moderate 10% increase in TPR. Comparing the multi-page classifier on the two settings, we see that it achieves a slightly higher TPR and a 22-fold reduction in FPR in the one-page setting. The significant reduction in FPR is likely more significant for most scenarios, i.e., the one-page setting is difficult mostly because having only one positive class drastically reduces FPR.

6.3 Notes on Deployment

The results of our work suggest that current defenses require a very large overhead increase to be effective in the one-page setting. While this means that general deployment against the one-page setting is likely too expensive, users who desire a higher level of privacy could still have the option to adopt it. Partial deployment of a WF defense can be done feasibly at low cost: as only the users who actively install and use the defense would incur a cost for the network, the overall burden on the network would remain low. Our higher-overhead modified Tamaraw can serve this purpose.

One may point out that incremental deployment can harm privacy as people who use the new version will be distinct from clients using the old version. This is an important consideration in e.g. browser fingerprinting and censorship resistance. Unlike these scenarios however, in WF, the attacker already knows the client's identity and only seeks to determine the client's behavior. The client's discernible willingness to adopt a WF defense only tells the attacker that the client cares about privacy, which we believe is not valuable information considering that the same client is already using Tor. It is nevertheless true that a larger anonymity set is beneficial for any privacy technology.

7 RELATED WORK

7.1 WF Defenses and What Broke Them

We give a brief overview of the history of WF defenses focusing on how they were broken.

Two early WF defenses, Adaptive Padding [14] and Traffic Morphing [20], were designed for HTTPS and VPN. The former focuses on covering interpacket timing and the latter on packet sizes. They were found to be ineffective against the first WF attacks that could attack Tor [3, 18]. In fact, some effective WF attacks do not use interpacket timing and/or packet sizes [11, 15].

Adaptive Padding was later modified and improved to become WTF-PAD [10]. It was able to show success against earlier WF attacks, but it was later broken with DF by Sirinam et al., based on Convolutional Neural Networks [15].

Two mimicry defenses were proposed, Supersequence [18] and Walkie-Talkie [19], but they both assume the client has some knowledge of the web pages to be loaded, and have not been proven practical to deploy. Tor implemented its own defense based on randomized pipelining in response to WF, but the defense has not proven effective against WF attacks and was removed during the upgrade to HTTP 2.0.

This work shows that three defenses, not broken in prior work, are not sufficiently strong in the one-page setting: Front, Decoy, and Tamaraw. Gong and Wang proposed Front [8] to cover the front of a packet sequence with dummy packets, as it is the most feature-rich portion of the sequence. Panchenko et al. proposed Decoy [12] to cover real page loads with fake page loads. Tamaraw [2] fixed a weakness of BuFLO [7], the first regularization defense, which was vulnerable as it did not cover packet sequences more than 10 seconds long.

7.2 Other defenses

The main goal of our work is to demonstrate the value of the one-page setting for the evaluation of WF defenses. To do so, we re-evaluated the best network-layer defenses, which constitute the majority of WF defenses. There are other defenses that can be applied to defeat WF as well, such as ALPaCA [4], which is a server-side defense; TrafficSliver [5], which aims to ensure that the attacker will only be able to see a small portion of the traffic[6]; and Glue [8], which adds dummy packets to glue together packet sequences

[6] TrafficSliver also investigates a slightly different attacker to our model: their attacker controls Tor nodes and would be made to see partial traffic under TrafficSliver, but our attacker is local to the client and will see all client traffic.

belonging to different web pages. As our focus on network-layer WF defenses suffices to show the importance of the one-page setting for WF defense evaluation, we did not tackle the difficult problem of implementing and comparing non-network defenses on the same basis as network-layer defenses, and we leave it as future work.

7.3 One-Page Setting in WF

We are not aware of any work that evaluated either attacks or defenses under the one-page setting — our results suggest that if those defenses were, they would have been seen as ineffective. The binary setting (two classes but with multiple monitored pages, as in Section 6.1) has sometimes been used to evaluate WF attacks in the open world [11, 12].

8 CONCLUSION AND FUTURE WORK

In this work, we set out to investigate WF defenses under the one-page setting. We found that several defenses, Front, Decoy, and Tamaraw, left the client vulnerable to WF attacks in the one-page setting. This was especially surprising for Tamaraw, which was designed as a future-proof defense against which any WF attack would fail. We found that in the one-page setting, the anonymity sets created by Tamaraw were too severely biased towards either class to be useful. Our investigation into bolstering these defenses shows that Tamaraw can become useful for the one-page setting with greater randomization at the cost of higher data overhead. We propose that the one-page setting should be used for all defense evaluation in the future.

We also explored a number of different WF scenarios that could not be captured by the standard laboratory scenario. We showed that WF attacks were indeed able to succeed in these scenarios in the one-page setting as well. These scenarios also introduced a number of new parameters that can significantly affect the attacker's performance. For the selection and identification scenarios, these include the number of total observations and the concentration of base rate. In the future, there may be more powerful attacks that can achieve success in these scenarios with few observations.

In our results, an improved version of Tamaraw is currently the best defense for the one-page setting, but it is not practical for large-scale deployment due to its high overhead costs and packet delays. One of its design flaws is a fixed constant packet rate, which is dissimilar to how real web pages are loaded; using varying packet rates that do not depend on the base page being loaded may be more efficient. Another possibility is that our pessimistic simulation may be over-estimating the cost of Tamaraw, and full evaluation on a real deployment may show better results.

We would like to thank the authors of the relevant works for sharing their code with us, as well as Gong and Wang for sharing their data set with us to allow our evaluation.

REFERENCES

[1] Sanjit Bhat, David Lu, Albert Kwon, and Srinivas Devadas. [n.d.]. Var-CNN: A Data-Efficient Website Fingerprinting Attack Based on Deep Learning. *Privacy Enhancing Technologies* 1, 19.

[2] Xiang Cai, Rishab Nithyanand, Tao Wang, Ian Goldberg, and Rob Johnson. 2014. A Systematic Approach to Developing and Evaluating Website Fingerprinting Defenses. In *Proceedings of the 21st ACM Conference on Computer and Communications Security.*

[3] Xiang Cai, Xin Cheng Zhang, Brijesh Joshi, and Rob Johnson. 2012. Touching from a Distance: Website Fingerprinting Attacks and Defenses. In *Proceedings of the 19th ACM Conference on Computer and Communications Security.* 605–616.

[4] Giovanni Cherubin, Jamie Hayes, and Marc Juarez. 2017. Website Fingerprinting Defenses at the Application Layer. *Proceedings on Privacy Enhancing Technologies* (2017).

[5] Wladimir De la Cadena, Asya Mitseva, Jens Hiller, Jan Pennekamp, Sebastian Reuter, Julian Filter, Thomas Engel, Klaus Wehrle, and Andriy Panchenko. 2020. TrafficSliver: Fighting Website Fingerprinting Attacks with Traffic Splitting. In *Proceedings of the 27th ACM Conference on Computer and Communications Security.*

[6] R. Dingledine, N. Mathewson, and P. Syverson. 2004. Tor: The second-generation onion router. In *Proceedings of the 13th USENIX Security Symposium.*

[7] Kevin P Dyer, Scott E Coull, Thomas Ristenpart, and Thomas Shrimpton. 2012. Peek-a-Boo, I Still See You: Why Efficient Traffic Analysis Countermeasures Fail. In *Proceedings of the 2012 IEEE Symposium on Security and Privacy.* 332–346.

[8] Jiajun Gong and Tao Wang. 2020. Zero-Delay Lightweight Defenses against Website Fingerprinting. In *Proceedings of the 29th USENIX Security Symposium (to appear).*

[9] Jamie Hayes and George Danezis. 2016. k-Fingerprinting: A Robust Scalable Website Fingerprinting Technique. In *Proceedings of the 25th USENIX Security Symposium.*

[10] Marc Juarez, Mohsen Imani, Mike Perry, Claudia Diaz, and Matthew Wright. 2016. Toward an Efficient Website Fingerprinting Defense. In *Computer Security–ESORICS 2016.* Springer, 27–46.

[11] Andriy Panchenko, Fabian Lanze, Andreas Zinnen, Martin Henze, Jan Pennekamp, Klaus Wehrle, and Thomas Engel. 2016. Website Fingerprinting at Internet Scale. In *Proceedings of the 23rd Network and Distributed System Security Symposium.*

[12] Andriy Panchenko, Lukas Niessen, Andreas Zinnen, and Thomas Engel. 2011. Website Fingerprinting in Onion Routing Based Anonymization Networks. In *Proceedings of the 10th ACM Workshop on Privacy in the Electronic Society.* 103–114.

[13] Vera Rimmer, Davy Preuveneers, Marc Juarez, Tom Van Goethem, and Wouter Joosen. 2018. Automated Website Fingerprinting through Deep Learning. In *Proceedings of the 25th Network and Distributed System Security Symposium.*

[14] Vitaly Shmatikov and Ming-Hsiu Wang. 2006. Timing analysis in low-latency mix networks: Attacks and defenses. In *Computer Security–ESORICS 2006.* 18–33.

[15] Payap Sirinam, Mohsen Imani, Marc Juarez, and Matthew Wright. 2018. Deep Fingerprinting: Undermining Website Fingerprinting Defenses with Deep Learning. In *Proceedings of the 25th ACM Conference on Computer and Communications Security.* ACM, 1928–1943.

[16] Qixiang Sun, Daniel R Simon, Yi-Min Wang, Wilf Russell, Venkata N Padmanabhan, and Lili Qiu. 2002. Statistical Identification of Encrypted Web Browsing Traffic. In *Proceedings of the 2002 IEEE Symposium on Security and Privacy.* IEEE, 19–30.

[17] Tao Wang. 2020. High Precision Open-World Website Fingerprinting. In *Proceedings of the 2020 IEEE Symposium on Security and Privacy.*

[18] Tao Wang, Xiang Cai, Rishab Nithyanand, Rob Johnson, and Ian Goldberg. 2014. Effective Attacks and Provable Defenses for Website Fingerprinting. In *Proceedings of the 23rd USENIX Security Symposium.*

[19] Tao Wang and Ian Goldberg. 2017. Walkie-Talkie: An Efficient Defense Against Passive Website Fingerprinting Attacks. In *Proceedings of the 26th USENIX Security Symposium.*

[20] Charles V Wright, Scott E Coull, and Fabian. Monrose. 2009. Traffic Morphing: An Efficient Defense against Statistical Traffic Analysis. In *Proceedings of the 16th Network and Distributed Security Symposium.* 237–250.