Evaluating the Security of HTTP/2 and HTTP/3 Against Website Fingerprinting Attacks

Yubo Shao

MInf Project (Part 1) Report
Master of Informatics
School of Informatics
University of Edinburgh

2024

Abstract

Website Fingerprinting (WF) attacks are causing significant concern about the users' confidentiality and privacy. Adversaries eavesdrop on victims and perform traffic analysis by passively collecting network features and using supervised learning techniques to reveal their web browsing behaviour, even if the victim is browsing in encrypted tunnels. However, it is challenging to have a formal evaluation of WF defences that fully captures their effectiveness using solely accuracy as the indicator to properly evaluate the security guarantee of a protocol. This paper aims to provide a detailed analysis of the security of HTTP/2 and HTTP/3 against WF attacks using the Bayes error lower bound technique.

Research Ethics Approval

This project was planned in accordance with the Informatics Research Ethics policy. It did not involve any aspects that required approval from the Informatics Research Ethics committee.

Declaration

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

(*Yubo Shao*)

Acknowledgements

My most sincere thanks to my supervisor Marc Juarez and my peers Max S. and Andy E. for providing valuable feedback and support throughout this project!

Table of Contents

Chapter 1

Introduction

1.1 Motivation

In today's day and age, where internet access has become an essential resource as important as food and shelter for many people, the confidentiality and privacy of internet traffic are crucial parts of any internet connection. Even with virtual private networks and Tor to protect the users and the destination server's IP to conceal information, eavesdropping and traffic analysis can still be performed using website fingerprinting (WF) attacks with extremely high accuracy without decryption or extensive privileges [13, 9]. It can be used to reveal an individual's browsing habits, track browsing history, build profiles, and cause severe damage to the general public.

Although many WF defences aid in hiding important network traffic features, we still don't have a definitive evaluation method to fully understand the extent of information leakage that WF attacks can cause. This further causes the actual security of transport protocols to be ambiguous.

Additionally, since GQUIC (Google Quick UDP Internet Connections) is standardized and integrated into "HTTP/3" [19] as IQUIC, the wide adoption of the new HTTP protocol since 2022 means that QUIC would also be the target of WF attacks and existing literature has found the QUIC protocols more vulnerable to WF attacks [19].

Hence, I am motivated to investigate and attempt to accurately evaluate the effectiveness of different WF attacks to properly evaluate the security of HTTP/2 and HTTP/3 against WF attacks. Also, by investigating the effect of different transport protocols on the performance of WF attacks, we can gain more insight into the inner workings of WF attacks and their real-world severity.

1.2 Aims and contributions

The goal of this project is to provide a detailed analysis of evaluating the security of HTTP/2 and HTTP/3 against WF attacks, focusing on the effect of different WF attacks used, using the Bayes error lower bound as the main evaluating factor proposed by

Cherubin [3].

I would like to demonstrate the following novel contributions in this paper:

- Created and trained models for WF attacks using WCN+ and self-collected traffic traces from HTTP/2 and HTTP/3, reproducing the results found by Cherubin [3].
- Comparisons and analysis of performance and behaviour of different WF attacks, including k-NN, CUMUL, LL, VNG++, and k-FP, using Bayes lower bound estimations on HTTP/2 and HTTP/3.
- Comparisons and analysis of performance and behaviour of different WF attacks against Tamaraw, a classic fixed-rate attack, using Bayes lower bound estimations on HTTP/2 and HTTP/3.
- Comparisons on the security guarantee of HTTP/2 and HTTP/3 protocols.
- Discussed the characteristics of realistically viable WF attacks on HTTP/2 and HTTP/3 indicated my the experimental results.

From this study, we identified the following key findings:

- HTTP/3 traffic has a marginally better security guarantee than HTTP/2 in full undefended traffic.
- HTTP/3 traffic has a significantly worse security guarantee than HTTP/2 in early undefended traffic ($10 \leq k \leq 70$).
- HTTP/3 traffic has a significantly worse security guarantee than HTTP/2 in full fixed-rate defended traffic.
- HTTP/3 traffic has a significantly worse security guarantee than HTTP/2 in early fixed-rate defended traffic ($30 \leq k \leq 50$).
- HTTP/3 is uniquely much weaker against burst traffic features than HTTP/2, exemplified by the behaviour of CUMUL against collected HTTP/2 and HTTP/3 traffic traces.

1.3 Project overview

The thesis begins with Chapter 2, a general introduction to the needed context on the process of WF attacks, QUIC and HTTPS protocols, existing defences, and existing defence evaluation methods. It also delivers a formalization of WF attacks, an estimation of the Bayes error and bounds on WF adversaries.

Chapter 3 details the experimental methodology including the data collection procedure and the evaluation method on both protocols. Chapter 4 provides an evaluation and analysis of the security guarantee indicated by the accuracy and the Bayes lower bound estimations of WF attacks for both transport protocols in a closed-world scenario. Chapter 5 completes the report with a summary of the findings, a discussion on the limitations of the experiments, additional work to be done, and potential future directions.

Chapter 2

Preliminaries

2.1 An informal conceptualization of Website Fingerprinting attacks

For any generated encrypted network traffic, although crucial private information such as the data payload and IP addresses are protected to an extent, information can still be extracted from the pattern, behaviour, and metadata associated with the traffic. In the case of Website Fingerprinting (WF) attacks, this information could be used by an adversary to train a classifier with a selected supervised learning algorithm to identify network traffic destinations and hence uncover the victim's browsing behaviour. This form of attack bypasses encryption protocols since there isn't any involvement with the actual messages or transmitted data but instead relies solely on the traffic itself.

To train the supervised learning model, general traffic traces need to be processed and converted to suitable data formats, such as different traffic features like the number of incoming and outgoing packets, unique packet sizes, and more. Additionally, each traffic trace is associated with a label that indicates the corresponding website and the adversary is aware of which traffic trace corresponds to which website. Different WF attacks utilise different sets of specific features as inputs and associated website labels as outputs to train their classifier using a chosen supervised learning algorithm. For example, *LL* attacks utilise the count of packets with a certain direction and size for each possible direction and size up to the maximum transmission unit and use the naive Bayes (NB) classifier for classification [11]. The feature set of *CUMUL* attacks includes information on packet sequences such as the total incoming and outgoing packets alongside the cumulative sum of packets' sizes, and *CUMUL* uses a Support Vector Machine (SVM) classifier for classification [14]. Hence, how revealing a feature set is and how well-protected the feature set is by the defender under different transport-layer protocols are significant factors in the effectiveness of a WF attack.

After training, the network traffic of a victim can be classified by the adversary using the trained model, and the generated traffic trace of the victim is treated as an unlabeled dataset. The targeted features in the utilised feature set of the attack are extracted from the generated traffic, and the model can predict the website that the victim is visiting.

The general procedure of common WF attacks is as follows:

1. Data Collection: The adversary collects traffic features by observation or uses existing datasets.
2. Feature Extraction & Data Labeling: A set of targeted features are extracted from the network traffic trace either by data pre-processing and fed into a classifier as inputs (ML-based attack) or by a feature-extracting layer contained in the deep-learning model (DL-based attack) [13]. The traffic traces associated with corresponding website labels are used as outputs.
3. Classifier training: The classifier is trained using a selected supervised learning algorithm.
4. Attack: Deploy the attack by eavesdropping on the victim's network traffic, and use the extracted feature set from the traces to predict websites visited by the victim using the trained model.

2.2 HTTPS and QUIC

The two most prominent transport-layer protocols in the Internet protocol suite are TCP (Transmission Control Protocol) and UDP (User Datagram Protocol). As the standard transport-layer protocol in HTTP/2, TCP provides a reliable connection with error detection and correction, which makes it the preferred protocol for applications that require reliable connections. However, existing issues like head-of-line blocking and high overhead leave performance to be desired. IETF attempted to resolve these issues in HTTP/3, where QUIC (Quick UDP Internet Connections) replaced TCP as the standard transport-layer protocol. The QUIC protocol operates over UDP and has taken the responsibility for multistream, encryption, congestion control, and reliable data stream from the combination of TLS and TCP altogether [19]. It solves the existing transmission performance shortfalls of TCP while achieving similar or better transmission efficiency as HTTPS in the majority of network conditions [19].

However, QUIC is not without its problems. Since QUIC is a TLS + TCP replacement and HTTP/2 suffers from WF attacks, many have done investigations on its ability to defend itself against WF attacks and found that it is more vulnerable than HTTP/2 in many cases, and feature sets that aren't as revealing in HTTP/2 became a lot more significant in the accuracy of the attack when using QUIC [19]. It is demonstrated by Zhan et al. [19] that, using their transfer feature set including unique packet size, packer order, inter-arrival time, and more, the feature importance is more concentrated when $k < 20$ (the first k packets in transmission) and relatively diffused on HTTPS even when $k < 50$, showing that fewer packets are required to achieve similar attack efficiency under QUIC as when attacking HTTP/2.

Considering these unique characteristics of QUIC, a proper evaluation of the performance of different WF attacks in HTTP/2 compared to HTTP/3 becomes important as it provides a measurable form of safety guarantee for both protocols.

2.3 Existing defences

To simplify a WF defence to its extreme, it effectively distorts the patterns in network traffic by introducing dummy packets, artificial delay, or traffic moulding to confuse the adversary's classifier with a reasonable bandwidth or latency overhead that does not affect the usability of the browser significantly.

For HTTPS and QUIC, some existing defences cannot be used since some are specifically designed with Tor in mind, like HyWF [7] and TrafficSliver [4]. Some have critical implementation issues, such as Walkie-Talkie [18, 16] and Mockingbird [15, 13]. Some heavily misrepresent the attacker's abilities and render themselves ineffective against state-of-the-art attacks such as DFD and BANP [13]. As of now, the following defences are particularly interesting in the setting of comparing WF attack performance between HTTP/2 and HTTP/3.

Fix-rate defences

The three well-known fix-rate defences are BuFLO [5], CS-BuFLO [1], and Tamaraw [2]. Fix-rate defences have high resistance against WF attacks as they effectively hide the traffic features by making all traffic appear uniform. However, this comes at a cost of very high bandwidth and latency overhead requirements. While these defences likely wouldn't be the most usable defences on HTTPS and QUIC, they would give a clear indication of the effect of WF defences against WF attacks to set a great baseline and to compare against the previous findings of the effectiveness of these defences on Tor by Cherubin [3].

DynaFlow

Where constant-rate defences send the traffic as a constant stream following a fixed pattern, DynaFlow [12] also adjusts the traffic rate of this constant traffic stream periodically using the average inter-packet arrival time from the previous period. DynaFlow is shown to be an effective defence against ML-based attacks and requires reasonable bandwidth and latency overheads. Its performance against DL-based attacks is further investigated by Mathews et al. [13], achieving 29% and 24% attacker accuracy when evaluating under the Safer and Safest security settings on Tor using the BigEnough dataset.

FRONT

As features produced from early traffic were significantly weighted by the classifiers shown by prior ML-based attacks, FRONT [6] targets the beginning of traffic traces by introducing fake packets with timestamps taken from two Rayleigh probability distributions produced dynamically for each trace. It performed well against ML and DL attacks with a maximum precision of 71% and a recall of 43% while having no latency and low bandwidth overhead requirements. The effect of FRONT on QUIC becomes interesting as Zhan et al. [19] found that QUIC's early traffic contains significantly revealing features and is more vulnerable to WF attacks compared to HTTPS.

2.4 Fingerprinting domain

One major assumption to be made before evaluating the effect of a WF attack is whether to use a close-world or open-world model.

The close-world model gives the adversary the most advantage and is usually used for stressing the WF defences or to provide an idealized environment for the adversary. Since in the close-world setting, the adversary has knowledge of a set of all accessible websites $W = \{w_1, w_2, ..., w_n\}$ so that all possible websites a user may visit are monitored. Assuming the probability of a user visiting any of the websites in W to be $1/n$, the adversary needs to identify the website that the user visited [10].

Comparatively, the adversary only knows a set of monitored websites in the open-world model. The goal for the adversary is to determine whether one of the monitored websites is visited and, if yes, which website. To approximate the user visiting a random non-monitored website, a set of non-monitored websites is introduced, and the probability of a user visiting a website is assumed to be based on the website's popularity regardless of whether it is monitored. This model is used to simulate real-world situations and evaluate a WF attack's capability in realistic environments [10].

Other attack models have also been used to simulate special cases or more realistic situations. In the "One vs. All" model, which is a special case of open-world [3], where the adversary only monitors one web page, the adversary only needs to determine whether the user is visiting a monitored page. A close-world model with browsing behaviour assumptions [8] was also proposed to simulate more realistic browsing behaviour in accordance with prior literature.

For the application of this paper, it is most sensible to use the close-world model to maximize the effectiveness of WF attacks to best evaluate the security of HTTP/2 and HTTP/3 against those attacks.

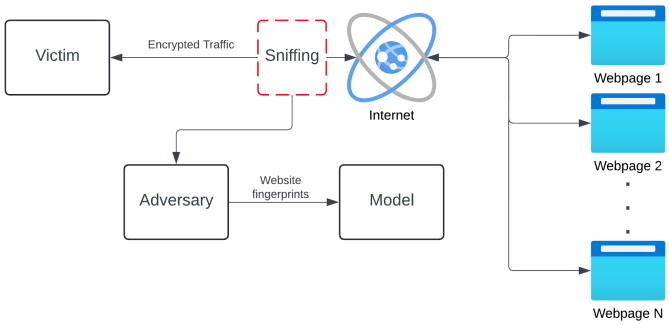
2.5 Threat model

With the fingerprinting domain set, the threat model of a WF adversary can be shown in Figure 2.1. The attacker first **passively** sniffing **encrypted traffic** generated by the victim visiting a range of webpages **locally** on the training stage. The attacker would typically operate in a local area network (LAN) or on the internet service provider (ISP) level and use the obtained data to generate website fingerprints with labels corresponding to each monitored webpage. The attack then uses the obtained website fingerprints to train a model.

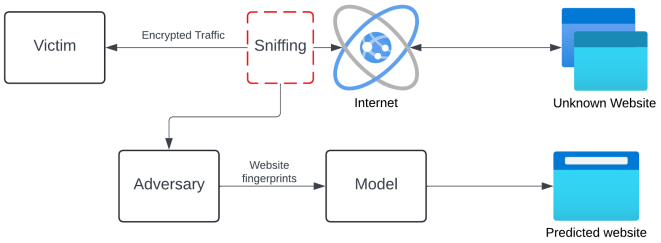
Since the fingerprinting domain is the closed-world model, all web pages that the victim visits are included in the monitored websites.

The deployment of the attack is done by first **passively** sniffing new **encrypted traffic** generated by the victim visiting the monitored websites **locally**. The website fingerprints obtained from the new traffic are fed to the model to give a prediction on the potential webpage the victim is visiting.

The attack model and website fingerprint will still reveal the website even if the plain material that can aid in identifying it cannot be directly sniffed from the traffic, harming the privacy that's provided by encryption.



(a) Training



(b) Attacking

Figure 2.1: Threat model of the website fingerprinting process

2.6 Accuracy: an inaccurate indicator?

In the experiment result analysis of the majority of existing papers, most evaluated the effectiveness of WF attacks and defences based on the accuracy of the state-of-the-art attacks and the overhead of defence for close-world scenarios. These indicators provided straightforward quantitative insight, but many argued that relying on accuracy alone is flawed for defence evaluation [10].

The accuracy of the attack is classifier-dependent; a good WF defence against a poorly trained classifier would result in low fingerprinting accuracy regardless. For example, in potential cases where the classifier accidentally reduced the set of likely web pages corresponding to a fingerprint due to noise or confusion but the classifier cannot reliably identify the correct page, it is not because of the lack of information provided by the fingerprint but instead caused by a poor classifier [10]. This also applies to a poorly chosen classifier for an effective feature set, the resulting low accuracy is not indicative of the real effectiveness of the feature set.

This shows that accuracy as an indicator tends to underestimate the effectiveness of the whole attack due to its classifier dependence and overestimate the performance of the WF defence.

2.7 Existing defence evaluation methods

Many have proposed provable security evaluation methods to more comprehensively capture the effectiveness of WF defences that do not rely solely on accuracy [2, 18, 10, 3]:

Cai et al. [2] proposed a method to calculate the lower bound of error achievable by a WF adversary by assuming an idealized adversary who has knowledge of a look-up table detailing *exactly* what packet sequence corresponds to what potential web pages. The smallest error achievable by an adversary would be the total collisions where different pages have the same packet sequence. This approach failed to consider the noise produced in network communications, which could result in misclassification for the adversary and underestimate the performance of some defences.

Wang et al. [18] proposed a probabilistic method to compute the probability distribution of defended network traffic for different web pages and to derive the smallest error achievable from the distribution. Unfortunately, the method assumed a look-up table adversary, which is greatly influenced by noise. Additionally, Cherubin [3] pointed out that the error is computed by running the defence a finite number of times, which only approximates the real distribution and doesn't guarantee a provable valid bound.

Cherubin [3] proposed the Bayes error lower bound technique, which resolves both issues of noise and approximating the real distribution. It doesn't assume a look-up table adversary, and, instead of running the defence a number of times to approximate the real probability distribution, the Bayes error lower bound (the smallest error achievable) is directly estimated and mathematically proven as the training set of size n approaches infinity.

Li et al. [10] took on a different approach and proposed the WeFDE (Website Fingerprint Density Estimation) method to estimate the mutual information shared between the information contained in the monitored websites' fingerprints and the distribution of these sites. This approach not only accounts for the hit-and-miss nature of accuracy as an indicator but near-hit and near-miss are also considered by evaluating based on total information leakage.

I elected to use the Bayes error lower bound approach considering it doesn't have the shortcomings of the aforementioned methods and there isn't meaningfully different information from the combination of accuracy and information leakage [13]. Additionally, some of its benefits are further elaborated in the following section.

2.8 The Bayes error

Cherubin [3] utilised the Bayes classifier in a WF defence evaluation context to produce the Bayes error and to further compute its lower bound. It measures the "smallest error achievable" as the overlapping area of the distributions between each of the collected features and their frequencies across the eavesdropped data for each of the web pages.

An intuition for the Bayes error is as follows (see Fig. 2.1):

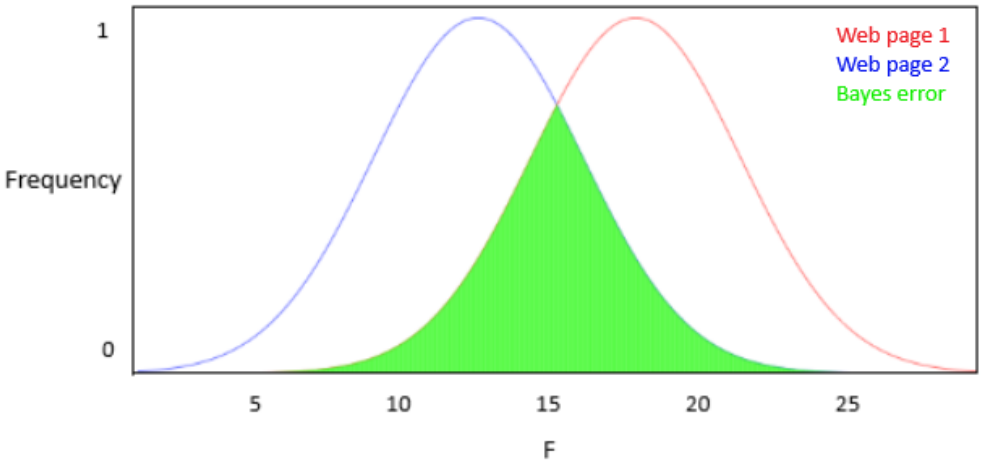


Figure 2.2: An intuitive example distribution for feature F and its frequency

- Assume there is only one feature F that can be observed by an adversary.
- Assume there are only two web pages.
- After observation, the probability distribution can be modelled between the feature F and the frequency of F being a certain value for each of the web pages.
- The Bayes classifier would predict the destination web page to be whichever web page has a higher frequency at any F value.
- For example, if feature F has a value of 10, the Bayes classifier would predict web page 2 as the correct one.
- The Bayes error is the smallest area where the prediction made by the Bayes classifier based on the value of F would be incorrect.
- For actual calculations, this process applies to an arbitrary number of web pages and features, and the Bayes error is computed based on the collected data.

Although the true distribution of different features would be unknown in practice and the true Bayes error cannot be known, the lower bounds of the Bayes error can be mathematically estimated to approach the real lower bound.

With this understanding, one of the biggest advantages of the Bayes error lower bound approach becomes apparent: It directly estimates the empirical Bayes error lower bound which is proven [3] to be lower than the actual Bayes error lower bound, which is lower than the theoretical Bayes error of the WF attack.

This means that the empirical lower bound of an attack is theoretically feature-set independent since no transformation of the original traffic trace (such as feature extraction) should improve the theoretical Bayes error. Also, this means that the estimated lower bound would always be below the empirical attack error rate regardless of the classifier choice. Hence, it acts as a great form of guarantee on the minimum performance of a WF attack.

2.9 Formulating ML and DL based adversaries

The purpose of this section is to provide a formulation of the adversaries in a closed-world setting, presenting a formal definition of attack error rate R^A, and showing that the definitions, proofs, and remarks for an ML-based adversary made by Cherubin [3] applies to both ML and DL based adversaries.

Similar notations from previous works [3, 2] are referenced and used.

W is a set containing all web pages that can be visited by a victim. Since we are under the close-world model assumption, all web pages are monitored by an adversary such that $|W| > 0$.

For a specific network traffic trace, a packet sequence is generated. Following the definition given by Cherubin [3], packet sequence $p \in P$ as a finite packet array of packet arrival time t_j, size s_j, and direction d_j:

$$p = ((t_j, s_j, d_j)) \text{ for } j = 1, 2, ...; \tag{2.1}$$

where $t_1 = 0$ and $t_j + 1 > t_j$, $s_j \in (0, MTU]$, and the direction $d_j \in \{\uparrow, \downarrow\}$.

For HTTPS which is using TCP, the MTU (maximum transmission unit) is 1500. The maximum QUIC packet is 1350 for IPv4 or 1330 for IPv6. Similar to Cai et al. [2], we make the same assumption that the packet sequence is the only obtainable information by a WF adversary when observing a traffic trace.

Using Cherubin's definition of label y and defence $D : P \to P$, from an ML-based adversary's standpoint, the features in the chosen feature set must be first extracted. For the set of feature extraction algorithms Φ:

$$\Phi = (\phi_1, \phi_2, ..., \phi_Q) \tag{2.2}$$

where $\phi_Q : P \to \mathbb{R}^{d_q}$ and $d_q > 0$ for $q = 1, 2, ..., Q$. Each ϕ_q is a feature extraction algorithm that takes the packet sequence as input and returns a vector of d_q real values. Each vector can contain a single value such as the total number of incoming packets, or some values such as packet inter-arrival time or transmission time.

Hence, applying the set of feature extraction algorithms on the traffic trace can be shown as applying Φ on P producing a fingerprint or features X of P:

$$\Phi : P \to X \tag{2.3}$$

where $X = \mathbb{R}^d$ and $d = \sum_{q=1}^{Q} d_q$. For a specifc traffic trace p:

$$\Phi(p) = (\phi_1(p), \phi_2(p), ..., \phi_Q(p)) = x \tag{2.4}$$

where x is the features extracted from p. In Cherubin's work [3], X is referred to as the object space and x is referred to as an object to disambiguate the definition of a feature.

In this case, the extraction algorithm ϕ and the feature itself x are distinctively separated and will be referred to separately.

An ML training algorithm $T_{ML} : (X \times Y)^* \mapsto F$ uses several fingerprint-label pairs as input to produce a classier $f \in F$ where $F = \{f \mid f : X \mapsto Y\}$. Hence, an ML-based WF adversary is a pair $A_{ML} = (\Phi,\ T_{ML})$.

The feature extraction step is not required for an adversary using a DL-based attack, hence the DL training algorithm $T_{DL} : (P \times Y)^* \mapsto M$ directly uses the traffic trace to produce a deep-learning model $m \in M$ where $M = \{m \mid m : P \mapsto Y\}$. We define A DL-based WF adversary as $A_{DL} = T_{DL}$.

Adapting from Cherubin's work [3], for an ML-based adversary $A_{ML} = (\Phi,\ T_{ML})$ given a defended traffic D, the classifier is trained using n pairs of packet sequences $((p'_i,\ y_i) = (D(p_i),\ y_i))$ for $i = 1,2,...,n$ and $y_i \in W$. Extracting the features for all i using Φ obtains the training set $Z_{train} = ((\Phi(p'_i),\ y_i)) = (x_i,\ y_i)$ and the classifier $f_{ML} = T_{ML}(Z_{train})$ is trained. When attacking, given a victim-generated packet sequence p'_{n+1} that has the actual label y_{n+1}, the adversary A_{ML} extracts features x_{n+1} and outputs prediction $f_{ML}(x_{n+1})$.

Hence, the rate of error $R^{A_{ML}} = P(f_{ML}(x_{n+1}) \neq y_{n+1})$ is the probability that the prediction doesn't match the actual label, which is one way of evaluating the performance of the adversary.

Extending the formulation to a DL-based adversary $A_{DL} = T_{DL}$, the classifier would be directly trained using the training set:

$$Z_{train} = (p'_i,\ y_i) \quad i = 1,2,...,n \quad y_i \in W \tag{2.5}$$

and obtain the classifier $f_{DL} = T_{DL}(Z_{train})$. In attacking scenarios, for packet sequence $(p'_{n+1},\ y_{n+1})$ the performance of the adversary can be formulated as $R^{A_{DL}} = P(f_{DL}(x_{n+1}) \neq y_{n+1})$.

Since the sole difference in the performance of adversaries R^A between ML and DL-based adversaries is the classier f itself, the proof provided by Cherubin regarding lower bounds estimates ($\hat{R}^*$) bounding the true lower bounds (R^*) and the lower bound's feature independence under full information remains valid for ML and DL-based adversaries [3]:

$$\hat{R}^* \leq R^* \leq R^A \tag{2.6}$$

Consequently, the formulations use the same assumptions made where the adversary trains on the one version of each webpage and there exists only one version of that webpage (standard i.i.d. assumption on all pairs of features and labels), and all labels are equally likely to be visited by the victim.

2.10 Related work

Mathews et al. [13] performed a comprehensive evaluation of different WF defences on Tor using the additional WeFDE technique under both close-world and open-world models. It further examined the DFD and BANP defences regarding their underestimation of the adversary's capabilities and evaluated the BiMorphing defence against deep-learning-based attacks, finding that Interspace, FRONT, and TrafficSliver appear to be the current best defences thus far in a Tor setting.

Cherubin [3] proposed the Bayes error lower bound technique and performed experimental analysis on fix-rate defences such as BuFLO, Tamaraw, and CS-BuFLO, as well as adaptive padding defences like WTF-PAD, against a variety of attacks, including state-of-the-art attacks such as ML-based attacks CUMUL and k-FP. Demonstrating that the WF adversary using a particular feature set is bound by the Bayes error.

Zhan et al. [19] investigated the effectiveness and characteristics of different WF attacks on QUIC traffic using two handcrafted feature sets: simple features and transfer features. They found that the early traffic of QUIC contains characteristic features that are extremely revealing compared to HTTPS.

Siby et al. [17] investigated the effectiveness of network-layer padding-based defences under the QUIC protocol in a close-world scenario. Their experiments using dummy packet injection based on FRONT have shown that padding-based network defences are ineffective against WF attacks despite their high overhead.

Chapter 3

Experimental Methodology

3.1 Data collection

There are a few requirements for the dataset to properly explore the differences between QUIC and HTTPS under WF attacks:

- Sufficiently large so that each trace contains enough information to extract features from.
- It needs to consist of encrypted traffic which simulates real-world situations of potential victims.
- It contains QUIC and HTTPS traffic originating only from the intended sources, cross origin resources would not be requested unless they are necessary to properly display the HTML.
- Collected in the same way as an adversary would.

Hence, a proper testbed setup is essential to simulate realistic environments to collect data.

The architecture of the testbed is shown in Figure 3.1. It used a modified version of the testbed architecture used by Zhan et al [19]. Since sniffing isn't done from the Internet directly and is instead done in a controlled network, it provides the ability to manage and regulate the traffic flow, which minimizes its impact on the network by not introducing unwanted individual functional packets to regular traffic. It also ensures the set of websites W visited by the victim all support both HTTP/2 and HTTP/3 and the traffic can be limited to HTTP/2 or HTTP/3 traffic only. Additionally, it helps with managing the traffic flow by preventing cross-domain resource requests and all generated traffic comes from a single source.

Similar to Zhan et al. [19], I selected the landing pages of the top 20 schools in the 2019 TIMES World University Rankings that support HTTP/3. I didn't use 100 pages, since the dataset format requirements for WCN+ don't demand 100 websites for it to be sufficiently effective at providing indicative trends[1], thus a smaller data set wouldn't

[1] `https://www.cs.sfu.ca/~taowang/wf/index.html`

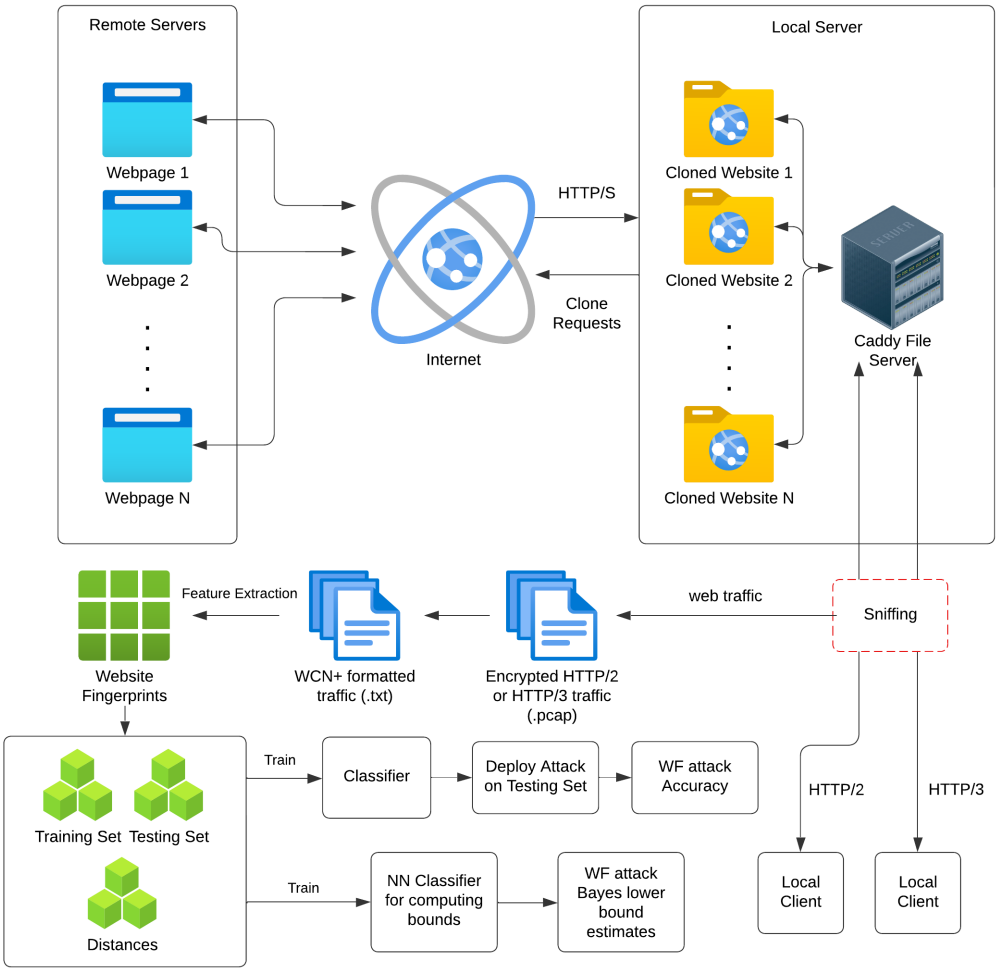


Figure 3.1: Overall Testbed Architecture

significantly impact the performance of WF attacks and it saves a significant amount of time in the web-crawling process.

I first cloned the website resources in the same origin with `wget` commands by enabling `--span-hosts` and `--page-requisites` to ensure that necessary files to display the landing pages are downloaded even if they are cross-origin, `--convert-links` to make them suitable for local viewing, and `--adjust-extension` to ensure the suffix `.html` to be appended to the local filename if the URL doesn't end with the regexp `\.[Hh][Tt][Mm][Ll]?`. I elected not to use HTTrack Website Copier[2] like Zhan et al. did as I found it fails to download some requirement cross-origin resources and limitations in customizations.

The web hosting server for the cloned websites is my personal machine with AMD® Ryzen 5 7600x, 16GB DDR5 RAM, and operating system Ubuntu 20.04.6 LTS. To

[2] `https://www.httrack.com/`

collect QUIC and HTTPS traffic, I used Caddy Server[3] in version 2.7.6. The local clients in the same local area network use Selenium version 4.17.2 with Chrome web driver version 121.0.6167.85 to automatically drive Chrome to simulate real-world user behaviour. The Chrome browser is used to visit the hosted websites with QUIC and HTTPS, as of the time of writing, the testbed works with any Chrome version from 121.0.6167.85 to 122.0.6261.69. To enable QUIC on Chrome, `--enable-quic` and `--origin-to-force-quic-on=<PORT>` is required. Since all web page visits are done locally, options `--headless` and `--no-sandbox` are used to speed up the visits, and a self-signed certificate for localhost is required to avoid `ERR_QUIC_PROTOCOL_ERROR` when visiting via HTTP/3. To prevent interference from previous visits and to further prevent cross-origin requests from being made, the cache and javascript should both be disabled.

Overall, a complete visit workflow is as follows:

1. Selenium begins a new Chrome process and the sniffing starts capturing localhost traffic.
2. Chrome attempts to visit the destination web page hosted locally and requests all necessary resources for the page to successfully load. Even if some external resource request fails, attempting to get the external resource is still a part of the signature of the webpage visit.
3. Chrome waits for the render completion of the destination page and returns its success state to Selenium.
4. Selenium sleeps for an arbitrary amount of time to ensure that all asynchronous operations on the page have been completed and then closes the current Chrome session.

For each webpage, there are 100 consecutive page loads as one full traffic capturing cycle, this is done separately for HTTPS and QUIC. `dumpcap` is used to sniff the local client's network interface (in this case, since both the server and client are hosted locally, the interface would be localhost) to imitate a realistic adversary. The captured traffic is stored in `.pcap` files, each file contains all captured traffic for one page-load of one website under either HTTP/2 or HTTP/3.

3.2 Evaluation framework

Evaluation is done using openly available code of major WF attacks and defences, and the public Python evaluation framework by Cherubin [3]. It is partially adapted and updated to suit the specific applications of evaluation regular HTTP/2 and HTTP/3 traffic instead of only Tor traffic.

The evaluation framework and available code of major WF attacks and defences accept data in the format of `WCN+` dataset, and my collected data is converted to the `WCN+` format to be used with these existing resources.

[3] `https://caddyserver.com/`

3.2.1 Evaluated attacks

The following attacks that focus on a range of both time and size features have their effectiveness explored:

LL

The Libertore and Levine (LL) website fingerprinting attack is a type of traffic analysis attack. It uses a feature set that primarily focuses on the size features including individual packet sizes, packet directions, and the count of packets with certain directions and sizes. The LL attack uses a Naive Bayes classifier to classify the traffic. However, although this attack is sensible on Tor traffic or traffic with some fixed packet sizes, its feature set is not very compatible with the nature of regular HTTP/2 and HTTP/3 traffic that have varying packet sizes. Additionally, finding a solution to the LL feature set and its further implementation is beyond the scope of this thesis. Hence, it is only used in result reproduction with WCN+ and not on collected HTTP/2 and HTTP/3 traffic.

VNG++

VNG++ is a website fingerprinting attack that uses a feature set of both time and size features including the bursts of packet sizes and directions (a sequence of neighbouring packets going in the same direction), the total time span of packets, and total per-direction bandwidth. It uses the Naive Bayes classifier. Similar to LL, some of its size features are designed for Tor traffic with an MTU of 1 rather than regular HTTP traffic with varying packet sizes. This makes VNG++ not particularly suitable for evaluation with HTTP/2 and HTTP/3 traffic. Considering it suffers from a similar problem as LL and adaptation would require effort out of the scope of the thesis, I elected to only use it in result reproduction with WCN+ and not on collected HTTP/2 and HTTP/3 traffic.

CUMUL

It uses a feature set that includes the cumulative sum of packet sizes besides the general features. This approach allows it to capture the burstiness of web traffic. The CUMUL attack uses a Support Vector Machine (SVM) classifier with an RNF kernel to classify the traffic and uses cross-validation grid search to determine the best parameters for the RBF kernel.

k-NN

Besides the basic features, it also uses transmission size, unique packet lengths, transposition, packet distributions (where are the outgoing packets concentrated) and burst features. A set of weights is determined for the features by their importance before classification and a modified version of the k-Nearest Neighbours classifier is used. The distance metric is Manhattan distance originally, but the Euclidean distance is used instead in the evaluation framework by Cherubin [3] for better performance.

k-FP

This is one of the state-of-the-art ML-based attacks. It uses a combination of time and size features based on the most effective ones in previous research analysis (a maximum of 175 can be used in one classifier when training) and it applies Random Forest (RF) to the feature values extracted from the original feature set and the generated leaves

are used for classification. It uses the modified k-NN classifier in the k-NN attack, originally Hamming distance is used as the distance metric but Euclidean distance is used in the evaluation framework by Cherubin [3] for better performance.

3.2.2 Evaluated defences

For *results reproduction*, the attacks are evaluated against two major fixed-rate defences:

CS-BuFLO

It is a modification of the well-known BuFLO defence, reducing BuFLO overheads and has a simpler implementation. It sends packets of fixed size s with frequency ρ similar to BuFLO, but rather than a fixed ρ, the value of ρ adapts dynamically to events like browser finishing loading a page or end of communication, as well as network bandwidth. Padding to the regular traffic is also added up to a fixed transmission size at the end of the communication [1].

Tamaraw

It also is a fixed-rate defence similar to BuFLO, but outgoing packets have frequency ρ_{out} and incoming traffic with ρ_{in} where $\rho_{out} > \rho_{in}$, where both frequencies are fixed. The packets sent in both directions have padding inserted based on a padding parameter [2].

For HTTP/2 and HTTP/3 traces, the attacks are evaluated against only Tamaraw, as adapting CS-BuFLO which is designed for defending Tor traffic requires substantial effort.

3.2.3 Dataset

Two datasets are used in the evaluation procedure:

The `WCN+` dataset [3] contains the time and size data of captured Tor traffic for 100 unique web pages with 90 page loads each. Since packet size is fixed at 1 in Tor, this lack of unique packet size sequence for web pages would result in the effectiveness decrease of size features.

The `WCN+` dataset is in the following format:

The filename is in the format of `$W-$L` where `$W` is the webpage number and `$L` is the page-load instance. The fifth page-load of webpage 1 would have filename `1-5`.

Every row of each file is in the format of `$T<tab>$S` where `$T` is the time when the packet is sent/received (time is 0 when traffic capturing begins), and `$S` is the size of the packet, positive size indicates outgoing traffic and negative indicates incoming traffic.

The other dataset is the data collected by me, which has 4000 `.pcap` files captured for both HTTP/2 and HTTP/3 in total, 2000 for each protocol ($n_{tot} = 2000$). For all packets in the captured traffic, testing has been done to guarantee that the packet size is not 0 or exceeding the MTU, the source and destination IP addresses are all expected, and all packets comply with the expected protocol.

Chapter 4

Evaluation

Scenario

The effectiveness of WF attacks and defences are evaluated from two different standpoints:

- The performance of each attack against the sizes of the total amount of data used in increments of 10% ($n = \{0.1 \times n_{tot}, 0.2 \times n_{tot}, ..., n_{tot}\}$). In all experiments that follows, $Z_{train} = 0.8 \times n$ and consequently $Z_{test} = 0.2 \times d$.
- The performance of each against the total amount of transmitted packets in increments of 10 packets ($k = 10, 20, ..., 70$) to examine their effectiveness on early traffic. It is important to note the k-FP attack requires at least 75 packets for their full performance and adaptation requires extensive effort, hence **k-FP is excluded in this case**.

Metric

The measurement for the performance of WF attacks is its error rate (i.e. attack failure rate) and the Bayes lower bound. Lower values for either metric are indicative of the attack's higher performance and hence worse security. Higher values for the Bayes lower bound indicate a better security guarantee for a protocol. The Bayes error lower bound estimations are obtained through 10-fold cross-validation (CV) unless specified otherwise and their standard deviations are shown in the figures. For the attack error rates, since they are obtained through a trained classifier, the classifier's predictions are deterministic based on its learned parameters and the output wouldn't change given the same input. Cross-validation is not used for the attack error rates.

4.1 Results reproduction with WCN+

I performed experiments in a Closed-world setting on the WCN+ data and evaluated the accuracy and lower bound of each attack against incremental sizes of n from $0.1 \times n_{tot}$ to n_{tot}. Following the same method of Cherubin [3], for an adversary $A = (\Phi, T)$, the empirical attack error $\hat{R}^A$ is compared against the computed lower bound $\hat{R}^*$ for the identical feature set Φ using 5-folds CV on the training set.

Figure 4.1 demonstrates the results of this experiment. The worst performing attacks are expectably older techniques such as LL and VNG++ with an attack error rate of 37.69% and 38.89% when $n = n_{tot}$ and relatively higher lower bound estimates compared to more modern techniques. Newer ML-based attacks such as k-NN and k-FP are pushing the boundaries of feature set effectiveness with attack error rates at 13.61% and 7.90% respectively when $n = n_{tot}$.

The lower bound estimates improve as the attacking technique becomes more recent: 18.06% for LL, 15.06% for VNG++, 7.28% for CUMUL, 6.12% for k-NN, and 7.22% for k-FP. The lower bound for k-NN is lower than k-FP while the bounds for CUMUL and k-FP are extremely close with CUMUL being slightly lower, these data show similar behaviour as the findings of Cherubin. The difference between the attack error rate and lower bound for each attack also decreases as the attack gets more recent, with a difference of 19.63% for LL, 7.49% for k-NN, and 0.41% for k-FP. This again indicates the potential limit to the raw performance of ML-based approaches using feature sets in recent years as Cherubin remarked.

Against defended traffic, there is a drastic increase in attack error rates and lower bounds, and the differences between the attack error rate and lower bounds widen significantly. This is expected from fixed-rate defences as all significant time and size sequences and signatures are destroyed to improve security at the cost of high bandwidth requirements and latency overhead.

CS-BuFLO has increased the lower bound LL, VNG++, CUMUL, and k-NN by a large margin, averaging 61.62% between the four, while k-FP performed well ahead of the rest, scoring 52.11% for its lower bound when $n = n_{tot}$. It's quite similar for accuracies as k-FP is performing at 57.48%, leaving a 5.37% gap from its maximum achievable performance while all other attacks have error rates above 80% and a gap above 20%, with k-NN performing the worst.

The situation is similar for Tamaraw as VNG++, CUMUL, k-NN, and k-FP grouped their lower bounds around 79% while LL's lower bound is at 90.53% when $n = n_{tot}$. However, Tamaraw successfully stopped k-FP as all attacks have their error rates above 90%, proving Tamaraw's better defence effectiveness over CS-BuFLO for Tor traffic.

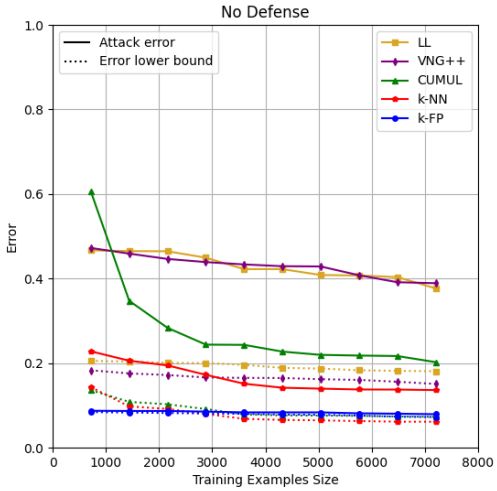
Overall, by examining the two heuristics used by Cherubin:

- the computed lower bound estimate of a feature set bounds the attack error rate of an attack using the identical feature set.
- the computed lower bound estimate should decrease as n increases.

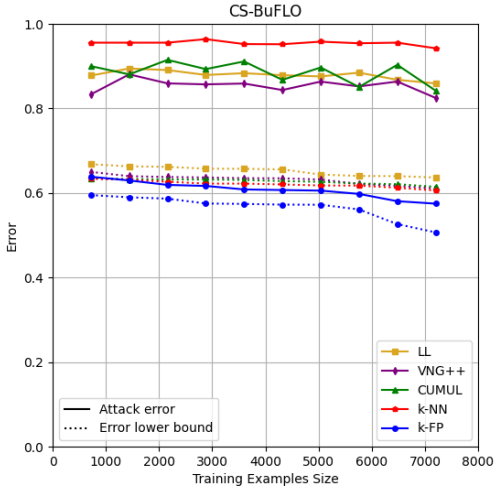
The reproduction results conclude that:

- For training set Z_{train} of any size, $\hat{R}^*$ for a feature set is lower than R^A of the same feature set.
- $\hat{R}^*$ decreases as n increases, and the decreasing differences between bound estimates and attack error rate for each feature set suggest potential convergence to an asymptote.

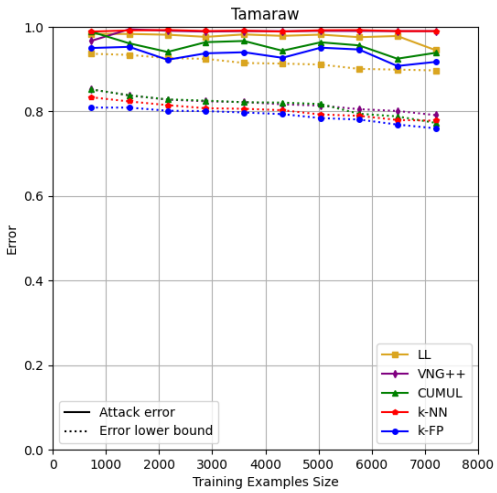
Which adheres to the findings of Cherubin.



(a) No defence



(b) CS-BuFLO



(c) Tamaraw

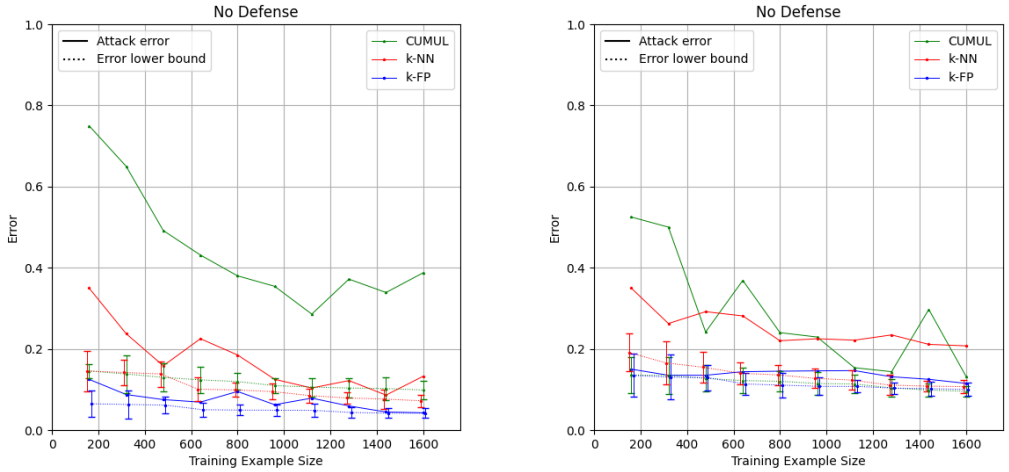
Figure 4.1: Lower bound $\hat{R}^*$ and attack error rate R^A on the WCN+ dataset (Closed-world) with respect to varying sizes of training examples Z_{train}.

4.2 WF attacks on HTTP/2 and HTTP/3

As shown by Figure 4.2, when performing the same experiment on HTTP/2 traces, the data shows a similar trend to WCN+ for lower bound estimates where it decreases as training examples size increases. Bounds for CUMUL, k-NN, and k-FP behave normally at 9.85%, 7.21%, and 4.15% respectively when $n = n_{tot}$.

For HTTP/3, CUMUL, k-NN, and k-FP have bounds at 9.54%, 10.70%, and 9.99% respectively when $n = n_{tot}$. Compared to HTTP/2, while the lower bounds for CUMUL are similar, the bounds are higher even when considering the standard deviation for k-NN and k-FP when $n \geq 0.3 \times n_{tot}$, albeit marginally. Additionally, the error rates for the three attacks are higher, implying there exist potential improvements to the classifying method for these feature sets on HTTP/3 relative to HTTP/2.

Figure 4.2 has also shown the advantage of Bayes lower bound estimations over evaluation purely from accuracy/error rate; The result error rate, which is affected by the choice and implementation of the classifier, can misrepresent the extent of the effectiveness or the information leakage of the feature set. In HTTP/2, while k-NN and k-FP are performing fairly close to their estimated lower bounds, CUMUL performed significantly worse looking at its error rate. This indicates the inability of the classifying method of CUMUL to sufficiently extract the information of the website signature from the transformed traces on HTTP/2.



(a) HTTP/2 (b) HTTP/3

Figure 4.2: Lower bound $\hat{R}^*$ and attack error rate $\hat{R}^A$ on collected HTTP/2 and HTTP/3 traces (Closed-world) with respect to varying sizes of training examples Z_{train}. (Lower bound markers are slightly offset to better present the standard deviations)

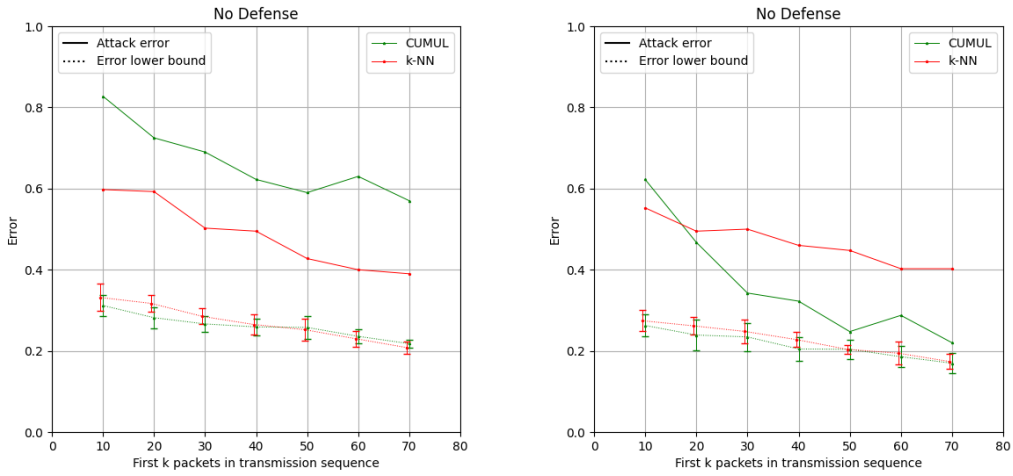
Compared to WCN+ traces, CUMUL performed significantly worse on HTTP protocols. This makes sense as rather than sending a series of continuous outgoing fixed-size packets on Tor, HTTP/2 and HTTP/3 would transmit larger-sized packets instead. Since burst features are an important aspect of CUMUL's feature set, it is penalized by this property of HTTP traces. CUMUL also performed substantially better on HTTP/3

compared to HTTP/2 while the bounds for HTTP/2 and HTTP/3 are quite similar, suggesting HTTP/3 traffic's weakness against burst features relative to HTTP/2.

The attack error rate of k-NN in HTTP/2 is generally within the standard deviations of the estimated lower bounds when $n \geq 0.6 \times n_{tot}$ indicating the classifier is properly extracting the effectiveness of the k-NN feature set. Comparatively, The attack error of k-NN in HTTP/3 is about 6.5% away from the standard deviations of its lower bound. This signifies how the change in transport protocol impacts the effectiveness and information leaked by an existing feature set negatively, and the same classifier is incapable of performing similarly well compared to its lower bound. This indicates the k-NN feature set's effectiveness may have decreased due to the different trace signatures of a different transport protocol.

Focusing on early traffic, as demonstrated by Figure 4.3, CUMUL and k-NN performed significantly better on HTTP/3 traffic than on HTTP/2 traffic in terms of error rates. On HTTP/2, CUMUL, and k-NN have respective error rates of 22.00% and 40.25%, and respective lower bounds of 16.96% and 17.38% when $k = 70$. Comparatively, the two attacks have respective error rates of 57.00% and 39.00%, and respective lower bounds of 21.78% and 20.81% when $k = 70$ on HTTP/2.

From the data, we can see the similar lower bounds of the two attacks across the range of early packets in both protocols, implying a similar effectiveness of the feature set of the CUMUL and k-NN on HTTP/2 and HTTP/3, even though k-NN has a substantially more elaborate and complicated feature set. However, the lower bounds on HTTP/3 are generally around 2.7% lower and are outside of the range of standard deviation. This indicates that HTTP/3 has a slight but definitely significant security guarantee deficit compared to HTTP/2 in early packets ($10 \leq k \leq 70$).



(a) HTTP/2

(b) HTTP/3

Figure 4.3: Lower bound $\hat{R}^*$ and attack error rate $\hat{R}^A$ on collected HTTP/2 and HTTP/3 traces (Closed-world) with respect to varying sizes of transmitted packets ($k \leq 70$). (Lower bound markers are slightly offset to better present the standard deviations)

However, one limitation to the assessment of ”HTTP/3 is less secure than HTTP/2” is that, while the collected dataset is above to provide indicative trends, its small size causes attacks to perform worse in terms of error rates and the lower bound estimations are less precise. This increases the loose of the bounds and even though the standard deviation suggests some significance, the actual difference may be too marginal to be definitively concluded.

Similarly to full traffic, CUMUL performs substantially better on HTTP/3 than HTTP/2 across k from 10 to 70 in terms of its error rate. It further suggests HTTP/3's weakness to CUMUL's burst-traffic-features-centric feature set, and it clearly demonstrates how the change in transport protocol affects the effectiveness of an existing feature set positively since the same classifier used by CUMUL is performing closer to its lower bound in HTTP/3 than HTTP/2. This indicates the CUMUL feature set may have its effectiveness increased by the different trace signatures of a different transport protocol.

This massive difference in CUMUL attack error rates between the two protocols again presents the importance of proper evaluation methods to the effectiveness of WF attacks, where the Bayes lower bounds provide a substantially less variable-dependent and relatively more credible way of security guarantee compared to using accuracy/error rates.

Overall, we found that for undefended traffic of HTTP/2 and HTTP/3:

- CUMUL performs better, indicating HTTP/3 particular weakness to CUMUL's feature set with a primary focus on burst traffic.
- For full traffic, HTTP/3 is marginally more secure than HTTP/2.
- For early traffic, HTTP/3 traces are more susceptible to existing classifying techniques and less secure than HTTP/2 by a small but significant margin.

4.3 Fixed-rate WF defences on HTTP/2 and HTTP/3

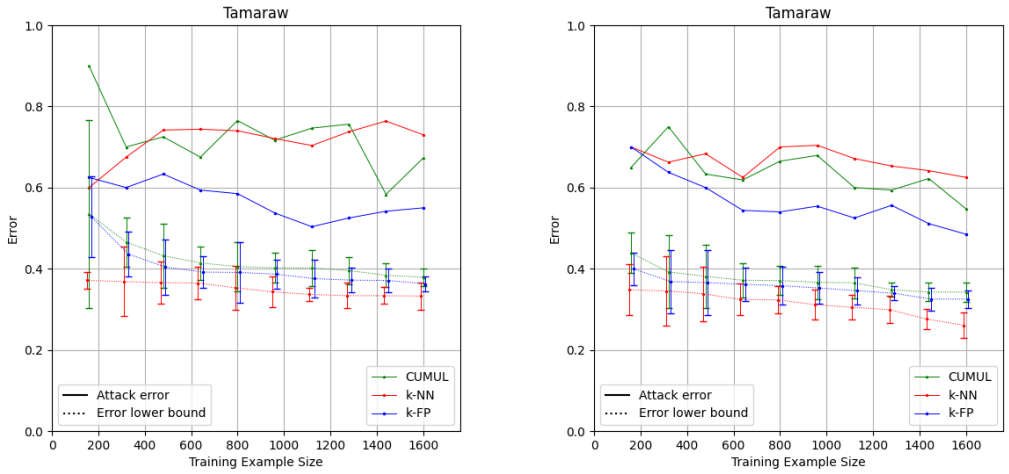
Against fixed-rate defended traffic, the majority of time and size trace signatures are destroyed, causing the attacks to be less performant across the training sizes as shown in Figure 4.4:

- The HTTP/2 lower bounds shifted to 37.91% for CUMUL, 33.26% for k-NN, and 36.25% for k-FP when $n = n_{tot}$, seeing an average increase of 28.42% between the four compared to undefended traffic.
- For HTTP/3, the lower bounds increased to 34.22%, 26.10% for k-NN, and 32.51% for k-FP, seeing an average increase of 21.67% between the four compared to undefended traffic.

The effectiveness of Tamaraw can be shown using the increase in the lower bounds of the three attacks compared to undefended traffic. From the obtained data, the lower bounds of attacks on HTTP/3 are below the HTTP/2 bounds across the board, indicating the feature sets used by the three attacks are more effective on HTTP/3 than HTTP/2. The utilisation of the Tamaraw defence to the undefended traffic increased the lower

bounds less for HTTP/3 than HTTP/2 as well. This indicates Tamaraw is less effective on HTTP/3 and hides less information compared to HTTP/2. Considering the similarity between Tamaraw and other major fixed-rate defences, this observation may indicate that the fixed-rate defences might perform worse in general on HTTP/3 relative to HTTP/2 in general. This also shows that HTTP/3 traffic is less secure than HTTP/2 traffic even when defended by a classic fixed-rate WF defense.

One interesting point shown by the data is that the difference between the error rates and the lower bounds is quite similar for all three attacks on both protocols. This signifies the importance of the feature set effectiveness for WF attacks, as then the majority of website signatures to extract features from are eliminated, the choice of training algorithms would not significantly impact the error rates nor the lower bounds of the attack. The similar standard deviations of attacks between HTTP/2 and HTTP/3 provide further evidence to the claim that the determining factor of a WF attack's effectiveness is its feature set, rather than its choice of training algorithms.



(a) HTTP/2

(b) HTTP/3

Figure 4.4: Lower bound $\hat{R}^*$ and attack error rate $\hat{R}^A$ on defended HTTP/2 and HTTP/3 traces (Closed-world) with respect to varying sizes of training examples Z_{train}. (Lower bound markers are slightly offset to better present the standard deviations)

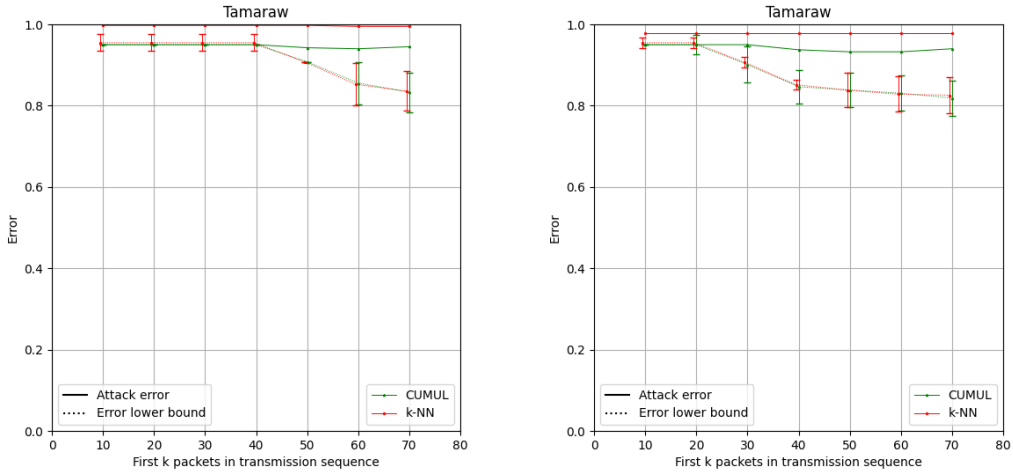
In terms of the differences in the error rates and their corresponding lower bounds:

- For HTTP/2, the gaps are 29.34% for CUMUL, 39.75% for k-NN, and 18.75% for k-FP when $k = 70$.
- For HTTP/3, the gaps are 20.42% for CUMUL, 36.42% for k-NN, and 15.99% for k-FP when $k = 70$.

These values show that even if Tamaraw has substantially penalized the feature set's capability, there are still more performance to be desired from the classifiers used for these attacks. Since the error rates are similar on both protocols, significant improvements potentially exist for all attack classifiers on fixed-rate defended traffic for both transport protocols.

For early defended traffic, the attacks behave quite similarly between the two protocols as shown in Figure 4.5. However, there exists clear albeit marginal differences:

The attack error rates for HTTP/3 are lower than HTTP/2 although very slightly. When $k = 70$, for HTTP/2, the error rate is 94.00% for CUMUL and 99.50% for k-NN; for HTTP/3 the error rate is 93.25% for CUMUL and 97.75% for k-NN. These are not significant enough to remark on the difference in classifier performance between the two protocols, but the behaviour of their respective attack lower bounds provides some interesting insights.



(a) HTTP/2 (b) HTTP/3

Figure 4.5: Lower bound $\hat{R}^*$ and attack error rate $\hat{R}^A$ on defended HTTP/2 and HTTP/3 traces (Closed-world) with respect to varying sizes of transmitted packets ($k \leq 70$). (Lower bound markers are slightly offset to better present the standard deviations)

The lower bounds for HTTP/2 and HTTP/3 when $k = 70$ have negligible differences, averaging a difference of 0.67% across four attacks and for $k \leq 20$, the differences are less than 0.01%. However, the data clearly shows that the lower bounds for HTTP/3 decrease significantly earlier at $k = 30$ whereas HTTP/2 bounds only start dropping until $k = 50$. There is an average 4.88% gap when $k = 30$ and an average 10.04% gap when $k = 40$ until the bounds start to converge at averaged 82.64% for both protocols.

This behaviour may be caused by the shorter handshake protocol implemented by QUIC, since a QUIC handshake requires only one round trip and a TCP + TLS handshake requires two round trips, QUIC usually establish connections and begin sending application data earlier than its HTTP/2 counterpart. Hence, more unique and identifiable metadata are present in earlier QUIC traffic compared to earlier TCP+TLS traffic and unique features can be extracted from QUIC traffic earlier using the feature set, thus explaining the earlier decrease of the lower bounds estimations on HTTP/3 compared to HTTP/2 traffic.

The minuscule difference between the lower bounds for both protocols implies that the majority of features are eliminated by Tamaraw. The plateauing lower bound estima-

tions are further evidence of this reasoning as after sufficient unique and identifiable metadata are transmitted ($k = 70$), the lower bounds for both protocols begin converging around 83%. This indicates a unique weakness of HTTP/3 and it is significantly less secure than HTTP/2 in early traffic $30 \leq k \leq 50$ and the error rates do not sufficiently reflect this difference, again, showing the limitations on evaluating attacks using solely accuracy/error rates.

Chapter 5

Conclusions and Dicussion

5.1 Summary

For this half of the project, the main goal of evaluating the security of HTTP/2 and HTTP/3 against WF attacks using the Bayes lower bound has been achieved.

I began work on the main goal by collecting traffic traces of different locally hosted websites using HTTP/2 and HTTP/3 and extracting various time and size features from the data.

I reproduced the results of Cherubin using the WCN+ dataset and confirmed his findings [3], then I conducted experiments on the difference in WF attack performance in terms of attack failure rates (error rates) and Bayes lower bounds against fixed-rate defended and undefended traffic using both HTTP/2 and HTTP/3.

Some of the scripts used during this project can be found here[1].

The results demonstrate:

- the lack of capability to properly represent aspects of WF attack evaluation;
- explored the Bayes lower bound estimation approach as a better alternative to accuracy as a WF attack evaluation methodology;
- the unique characteristics of HTTP/2 and HTTP/3 against WF attacks;
- the effectiveness of Tamaraw against WF attacks when defending HTTP/2 and HTTP/3 traces;
- the effect of different training example sizes on the error rate and Bayes lower bound estimation of a WF attack;
- the effect of first k packets on the error rate and the Bayes lower bound estimation of a WF attack.

[1] `https://github.com/yuboshaouoe/websitefingerprinting-transport-protocol.git`

5.2 Discussions

Efficiency of WF attacks on HTTP/2 and HTTP/3

While CUMUL's attack error rates are not as good as k-NN and k-FP on both undefended and defended full traffic traces, it has a relatively close lower bound to those more elaborate attacks while being significantly less resource-intensive to implement and run. Qualitatively, CUMUL completes feature extraction, training, classification, and bound calculation in significantly less time compared to k-NN and k-FP during the experiment procedure. It also handles larger datasets better than k-NN and k-FP due to its SVM classifier. A hypothesis can be made that CUMUL is a generally good practical option for attacking HTTP/3 traffic considering its performance to efficiency balance.

This comes with several caveats:

- The dataset used in the conducted experiments is fairly small and can lead to an increased looseness on the Bayes lower bound estimations;
- The distance between the empirical Bayes lower bound of CUMUL and its actual theoretical bound is unknown;
- The default hyperparameters are used in the conducted experiments, additional tuning may further worsen or improve the estimated lower bounds of other attacks.

Other aspects of realistically implementing Tamaraw

Realistically implementing Tamaraw on HTTP/2 and HTTP/3, while it is very effective against WF attacks, has some significant impact on the user experience and performance loss:

- For the collected HTTP/2 data, defending the traces with Tamaraw caused a 240.0% volume overhead and a 3708.0% time overhead.
- For the collected HTTP/3 data, defending the traces with Tamaraw caused a 217.0% volume overhead and a 5000.0% time overhead.

These overheads are too substantial to be implemented for HTTP/2 and HTTP/3 traffic under a practical setting. However, it is a good benchmark for a WF defence in the setting of the conducted experiments.

5.3 Limitaions

I identify that there exist several areas in which the experiments and further analyses can be improved upon, including:

- The lack of different types of WF defences, especially WF defences that focus on the specific weaknesses of HTTP/3 such as FRONT[6];
- The lack of different types of WF attacks, especially deep-learning-based WF attacks doesn't rely feature-extracting algorithms;

- A relatively small data set that causes the estimations to be less precise and lower bounds estimations to be looser;
- While the Bayes error lower bound estimations bound the actual theoretical Bayes error lower bound, the distance between them is unknown;
- The lack of hyperparameter tuning, the conducted experiment only used the default values.

5.4 Future work

I recognize the existing limitations to my experiment results and will continue to improve on them in the second half of this project: potentially using larger and more comprehensive datasets, implementing a wider range of state-of-the-art attacks and defences, and other alternative evaluation methods than accuracy and Bayes error lower bound estimation.

Bibliography

[1] Xiang Cai, Rishab Nithyanand, and Rob Johnson. Cs-buflo: A congestion sensitive website fingerprinting defense. In *Proceedings of the 13th Workshop on Privacy in the Electronic Society*, WPES '14, page 121–130, New York, NY, USA, 2014. Association for Computing Machinery.

[2] Xiang Cai, Rishab Nithyanand, Tao Wang, Rob Johnson, and Ian Goldberg. A systematic approach to developing and evaluating website fingerprinting defenses. In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*, CCS '14, page 227–238, New York, NY, USA, 2014. Association for Computing Machinery.

[3] Giovanni Cherubin. Bayes, not naïve: Security bounds on website fingerprinting defenses. *Proceedings on Privacy Enhancing Technologies*, 2017(4):215–231, 2017.

[4] Wladimir De la Cadena, Asya Mitseva, Jens Hiller, Jan Pennekamp, Sebastian Reuter, Julian Filter, Thomas Engel, Klaus Wehrle, and Andriy Panchenko. Trafficsliver: Fighting website fingerprinting attacks with traffic splitting. In *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*, CCS '20, page 1971–1985, New York, NY, USA, 2020. Association for Computing Machinery.

[5] Kevin P. Dyer, Scott E. Coull, Thomas Ristenpart, and Thomas Shrimpton. Peek-a-boo, i still see you: Why efficient traffic analysis countermeasures fail. In *2012 IEEE Symposium on Security and Privacy*, pages 332–346, 2012.

[6] Jiajun Gong and Tao Wang. Zero-delay lightweight defenses against website fingerprinting. In *Proceedings of the 29th USENIX Conference on Security Symposium*, SEC'20, USA, 2020. USENIX Association.

[7] Sébastien Henri, Gines Garcia-Aviles, Pablo Serrano, Albert Banchs, and Patrick Thiran. Protecting against website fingerprinting with multihoming. *Proceedings on Privacy Enhancing Technologies*, 2020:89 – 110, 2020.

[8] Marc Juarez, Sadia Afroz, Gunes Acar, Claudia Diaz, and Rachel Greenstadt. A critical evaluation of website fingerprinting attacks. In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*, CCS '14, page 263–274, New York, NY, USA, 2014. Association for Computing Machinery.

[9] Marc Juarez, Mohsen Imani, Mike Perry, Claudia Díaz, and Matthew Wright.

WTF-PAD: toward an efficient website fingerprinting defense for tor. *CoRR*, abs/1512.00524, 2015.

[10] Shuai Li, Huajun Guo, and Nicholas Hopper. Measuring information leakage in website fingerprinting attacks and defenses. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, CCS '18, page 1977–1992, New York, NY, USA, 2018. Association for Computing Machinery.

[11] Marc Liberatore and Brian Neil Levine. Inferring the source of encrypted http connections. In *Proceedings of the 13th ACM Conference on Computer and Communications Security*, CCS '06, page 255–263, New York, NY, USA, 2006. Association for Computing Machinery.

[12] David Lu, Sanjit Bhat, Albert Kwon, and Srinivas Devadas. Dynaflow: An efficient website fingerprinting defense based on dynamically-adjusting flows. In *Proceedings of the 2018 Workshop on Privacy in the Electronic Society*, WPES'18, page 109–113, New York, NY, USA, 2018. Association for Computing Machinery.

[13] Nate Mathews, James K Holland, Se Eun Oh, Mohammad Saidur Rahman, Nicholas Hopper, and Matthew Wright. Sok: A critical evaluation of efficient website fingerprinting defenses. In *2023 IEEE Symposium on Security and Privacy (SP)*, pages 969–986, 2023.

[14] Andriy Panchenko, Fabian Lanze, Jan Pennekamp, Thomas Engel, Andreas Zinnen, Martin Henze, and Klaus Wehrle. Website fingerprinting at internet scale. In *23rd Annual Network and Distributed System Security Symposium, NDSS 2016, San Diego, California, USA, February 21-24, 2016*. The Internet Society, 2016.

[15] Mohammad Saidur Rahman, Mohsen Imani, Nate Mathews, and Matthew Wright. Mockingbird: Defending against deep-learning-based website fingerprinting attacks with adversarial traces, 2020.

[16] Mohammad Saidur Rahman, Payap Sirinam, Nate Mathews, Kantha Girish Gangadhara, and Matthew Wright. Tik-tok : The utility of packet timing in website fingerprinting attacks. *Proceedings on Privacy Enhancing Technologies*, 2020(3).

[17] Sandra Siby, Ludovic Barman, Christopher Wood, Marwan Fayed, Nick Sullivan, and Carmela Troncoso. You get padding, everybody gets padding! you get privacy? evaluating practical quic website fingerprinting protections for the masses, 2022.

[18] Tao Wang and Ian Goldberg. Walkie-Talkie: An efficient defense against passive website fingerprinting attacks. In *26th USENIX Security Symposium (USENIX Security 17)*, pages 1375–1390, Vancouver, BC, August 2017. USENIX Association.

[19] Pengwei Zhan, Liming Wang, and Yi Tang. Website fingerprinting on early quic traffic. *Computer Networks*, 200:108538, 2021.