DNS-over-QUIC and HTTP/3 in the Era of Transformers: The New Internet Privacy Battle

Levente Csikor, Ziyue Lian, Haoran Zhang, Nitya Lakshmanan, Dinil Mon Divakaran

***Abstract*—Moving away from plain-text DNS communications, users now can switch to encrypted DNS protocols for name resolutions. DNS-over-QUIC (DoQ) employs QUIC—the latest transport protocol—for encrypted communications between users and their recursive DNS servers. QUIC is also poised to become the foundation of our daily web browsing by becoming the transport for HTTP/3, the latest version of the HTTP protocol.**

Traditional TCP-based web browsing is vulnerable to website fingerprinting (WFP) attacks that can identify the websites a user visits. The emergence of QUIC-based DNS and HTTP protocols raises an important question: are regular users better protected from WFP attacks when using these new protocols?

To investigate this, we first collect and publicly release the first benchmark dataset of network traffic corresponding to real visits to QUIC-enabled websites while using DoQ for domain resolution. This dataset will help advance the research on WFP attacks and defenses. Second, we implement and evaluate the first WFP attack targeting the combined use of DoQ and HTTP/3 protocols by users by developing two transformer models tailored for WFP attacks. Finally, we conduct comprehensive experiments, which reveal that these models are effective in identifying user-visited websites, emphasizing the need for defensive measures.

I. Introduction

In today's evolving digital landscape, protecting user privacy is paramount as well as challenging. Among various threats that exist, a man-in-the-middle (MitM) attacker with access to a network middlebox, such as a router, can intercept and read the transiting packets, thereby potentially compromising a user's privacy. In a website fingerprinting (WFP) attack, the attacker aims to identify sensitive websites visited by targeted users by eavesdropping on the underlying network communications, consequently leading to the leak of private information. Such leak could be exploited to discern an individual's sensitive information [1], e.g., political views, health conditions, and product/banking preferences. Profiling can subsequently lead to other threats (e.g., information sold in the dark market), attacks, and even censorship [2].

Although much of our web browsing traffic is encrypted due to the wide adoption of TLS, the DNS (Domain Name System) protocol responsible for domain name resolutions, has historically operated as a plain-text protocol. Since virtually every online communication necessitates DNS resolution of one or more domains, the plain-text queries and responses of the DNS protocol serve as attack vectors for malicious actors to launch multiple attacks on users, including WFP attacks.

In response to potential threats, standardized encrypted DNS protocols such as DoT (DNS-over-TLS, RFC 7858) and DoH (DNS-over-HTTPS, RFC 8484) have seen increasing adoption across various operating systems and browsers. In particular, the rise of the latest encrypted DNS protocol, DNS-over-QUIC (DoQ), and the shift in web browsing towards HTTP/3 (which is over QUIC, too) are transforming network communications.

Network protocols: As web communications move to the next standard of HTTP/3 (H3, RFC 9114), TCP protocol is being replaced by QUIC (RFC 9000), promising improved latency, throughput, resilience to client mobility, security, and privacy. In particular, QUIC eliminates TCP's head-of-line blocking [3], and reduces connection setup time with built-in TLS encryption, significantly reducing the time-to-first-byte. The recently standardized DNS-over-QUIC (DoQ) integrates these advantages with encrypted DNS, outperforming DoT and DoH in web performance [4]. Despite being relatively new, DoQ is already being adopted by major DNS providers like AdGuard and NextDNS.

While protocols like TLS ensure data confidentiality, there has been a rise in machine learning-based WFP attacks that threaten user privacy. Recently, these models have become more powerful and easier to train, enhancing the capabilities of attackers.

Advancements in AI: AI is progressing rapidly, largely driven by the transformer model [5], which has surpassed previous deep learning (DL) sequence models like recurrent neural networks (RNNs) in efficiently processing long data sequences and understanding context. Understanding how changes in network protocols, combined with recent advancements in AI, impact user vulnerability to WFP attacks is now essential for safeguarding online privacy and developing effective countermeasures.

Consequently, a pressing question arises:

> Are regular users safe from WFP attacks when using the latest QUIC protocol for *both* DNS and web browsing? Or are they still at risk due to the advanced AI models?

Our research work addresses exactly this question (see Threat Model in Sec. III). Different from existing works, to the best of our knowledge, this is the first research to analyze WFP attacks in the context of both DoQ and HTTP/3 (that runs on top of QUIC). Internet communications is expected to transition to these two recent and important protocols. With DoQ still in its nascent stage and QUIC (or HTTP/3) yet to be enabled widely on the top 1 million websites (see Sec. IV-B), we deem it timely to evaluate these emerging protocols from the perspective of WFP attacks.

Additionally, we introduce the first benchmark dataset of DoQ and HTTP/3 traffic sourced from visits to popular QUIC-enabled websites, along with Docker scripts, to facilitate reproducible and systematic data collection (Sec. IV). As detailed

later, the network traffic is generated from real web browsing sessions that (automatically) use DoQ for name resolutions and HTTP/3 for accessing well-known websites. Our data collection process ensured that all DNS communications were exclusively carried over QUIC (i.e., DoQ), further aligning with the protocol's deployment. To enhance the quality and comprehensiveness of the dataset, the data was collected from multiple vantage points to better capture the network characteristics associated with these emerging protocols.

To evaluate the utility of our dataset, we implement the first WFP attack on DoQ and HTTP/3 traffic (Sec. V). We develop two deep learning models based on the transformer architecture: one uses only DoQ traffic; while the other incorporates packets from a browsing session consisting of DoQ for domain resolution, and both QUIC and TCP for transporting web traffic. The models are trained using meta-information of encrypted packets to recognize patterns of different website traffic. The extensive evaluations (Sec. VI) show that the transformer models achieve over 70% recall at 90% precision, highlighting the capability of a WFP attack to effectively identify the websites that a user visits.

II. Background and Related Works

A. From TCP to QUIC

For decades, TCP has been the primary choice for web communications, providing reliable and ordered delivery of packets over the internet. However, the three-way handshake and head-of-line blocking in TCP [3] at the transport layer hurt latency. This affects applications like web communication (HTTP). Even with multiplexing strategies at the application layer, such as HTTP/1.1 vs HTTP/2, connections remain inefficient. While several solutions for TCP are being proposed to address these, e.g., TCP Fast Open, Google took steps that led to the development of an entirely new protocol, QUIC.

QUIC is a user-space transport protocol running on top of UDP, giving more control to applications. Several research studies examine the pros and cons of QUIC. Yet, it has notable advantages compared to TCP. First, with inherent support for TLS, QUIC achieves faster connection establishment by incurring fewer round-trips than TCP along with TLS. QUIC establishes a full connection in 1-RTT and connection resumption in 0-RTT. Second, QUIC multiplexes *streams*, which makes a stream independent of packet losses in other streams, thus overcoming the long-standing head-of-line blocking problem in TCP [3]. This has implications for different applications, and more so for web browsing, since modern web pages have multiple components, making transporting using QUIC an option for faster page loads. HTTP/3 (H3), the latest version of the web browsing protocol, runs on QUIC. While QUIC is a relatively new protocol (first deployed in 2013), it is now used by more than 32% of all websites[1].

B. Plain-text DNS to encrypted DNS protocols

Since DNS was created, its plain-text design has allowed attackers to exploit it in different ways. For example, attackers can change DNS resolutions for various services and send users to fake websites. The lack of encryption also allows attackers to track and block users' online activities. To solve this, DNS-over-TLS (DoT) was introduced in 2015. It encrypts DNS traffic between clients and resolvers. Then, DNS-over-HTTPS (DoH) was introduced in 2017. Both use TCP and provide the same encryption through TLS.

However, in 2022, the IETF standardized DNS-over-QUIC (DoQ, RFC 9250) to optimize privacy by minimizing the latency. With the implementation of DoQ, connections are established faster than DoT/DoH. Moreover, QUIC provides additional encryption options, making DoQ competitive with DoH in terms of speed, packet loss rates, and encryption capabilities. The use of DoQ has been observed to load web pages 10% faster in comparison to DoH [4]. Interestingly, it was also observed that, as the number of domain resolutions required increases for a complex website, the page load time becomes only around 2% slower with DoQ than with the traditional plain-text DNS (over UDP) [4]. This marginal performance impact is attributed to the amortization of encryption costs across multiple domain resolutions.

C. Website fingerprinting over the years

Website fingerprinting [1], [6] has been studied extensively over the past two decades. Most research considers TCP as the underlying communication for both encrypted HTTP and DNS communication. While in certain settings, side channels, such as plain-text DNS, and SNI in TLS, might leak information regarding the websites being visited, the common assumption in WFP studies is that no such information is available. The major focus of the existing WFP works has been on i) defining network-level features for training the WFP models and ii) adapting the latest machine learning models to achieve high accuracy.

Features: Since the traffic is encrypted, the features used are meta-information from the packet headers (and not payloads). These features fall into two types:

- Raw per-packet features: These are extracted directly from the packet headers, such as packet size, inter-arrival time, and direction [1], [7], [8].
- Engineered (or hand-crafted) features: These features are created by grouping packets based on certain rules. For example, a burst of packets with a short inter-arrival time is grouped together. Features such as total size, the number/ratio of incoming and outgoing packets, and statistical measures of inter-arrival times (e.g., mean and standard deviation) are then computed and used.

Models: Several research works study the effectiveness of conventional models, e.g., Naive Bayes classifier, SVM (support vector machine), and Random Forest, for building the WFP attacker system. The advancement in neural networks led to the development of deep learning models that can learn more from not only large datasets but also from high-dimensional features. In the WFP domain, this resulted in proposals that leveraged AutoEncoders (AE), CNN, LSTM and a combination of multiple models [6]. AEs, for example,

[1]https://w3techs.com/technologies/details/ce-http3

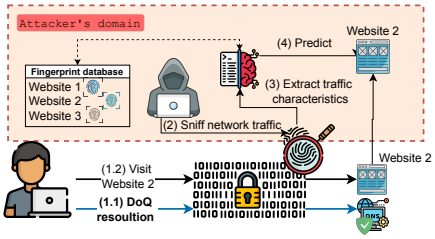


Fig. 1. Threat model of WFP attacks.

are useful in encoding features from network packets, which can then be fed to a state-of-the-art DL classifier [6].

QUIC website fingerprinting: As websites began using QUIC, recent research analyzes traffic to see if QUIC sites are vulnerable to WFP attacks. In [9], the authors show that a model trained on TCP traffic struggles to identify QUIC websites. However, using several engineered features, the model can identify QUIC traffic for website fingerprinting.

Encrypted DNS: The standardization of encrypted DNS protocols led to new studies on traffic analysis for security and privacy [10]. For example, with DoH, since it uses the same port as HTTPS (443), DNS and web traffic *cannot* be distinguished by applying rules on network traffic. However, in [11], we showed that machine learning models, using raw features like packet size and inter-arrival time, can identify DoH traffic, making DoH fingerprinting attacks possible.

To the best of our knowledge, no existing research work studies DoQ in the context of fingerprinting websites that support QUIC. Below, after defining the threat model, we define the problem we are addressing in this work.

III. Threat Model

Our threat model, depicted in Fig. 1, follows the traditional approach in WFP attacks. We consider a network attacker with the ability to passively monitor the network traffic communications of a target victim. Following common assumptions in WFP studies [12], this attacker may control an intermediate network device, such as a router, as seen in authoritarian regimes and malicious ISPs that monitor their citizens and customers [2]. Alternatively, they could also monitor users through a compromised router or WiFi hotspot. With this eavesdropping capability, the attacker intercepts and logs packets in transit between users and the remote services they access (including websites and DNS servers). In this work, we assume that the communication protocols utilized by users for browsing, as well as by attackers for fingerprinting, are *(i)* DoQ for DNS resolution and *(ii)* QUIC for web browsing. Note that websites supporting QUIC might also generate TCP flows, for instance, to deliver part of the contents from other resources (e.g., content distribution networks) that do not yet support QUIC.

To execute the attack, the adversary employs a supervised machine learning model. The dataset used for model training is generated and collected by the adversary beforehand. Specifically, the adversary visits a set of sensitive websites (i.e., monitored websites) of interest and captures the corresponding packet traces. Additionally, the adversary collects network traces of websites that are not sensitive (i.e., unmonitored websites). This enables the attacker to train a model to differentiate between monitored and unmonitored websites in a realistic scenario; refer to Sec. VI-B for a description of this *open world* setting.

A. Problem Definition

Taking the attacker's role, we are interested in identifying the QUIC-supported websites that a user visits. For this, we formulate the problem in two scenarios:

- $\mathcal{S}^{\text{DoQ}}$: Can an attacker identify the websites visited by a user solely relying on DoQ traffic? To answer this question, we focus on the DoQ traffic related to a website's domain resolutions and its resources. We train a model to classify and identify the monitored websites. Observe that, domain resolutions make up only a small part of the traffic when visiting a website; most of the traffic is the website's content. This limited data makes the problem challenging. However, it also helps us understand the privacy risks caused by DoQ alone.
- $\mathcal{S}^{\text{DoQ+H3}}$: We train a website classification model using the first k network packets generated during visiting a QUIC-supported website. Importantly, this sequence of packets includes DoQ packets besides the web traffic, and, as mentioned above, the latter not only consists of QUIC but also TCP packets [9].

IV. A New Dataset of DoQ and QUIC Traffic

Given that DoQ and QUIC are relatively new protocols, there is no readily available datasets for our work. We describe the process of collecting network traffic for DoQ and QUIC.

A. Connecting to Website via QUIC

Since QUIC is a young protocol, not all websites have implemented support for it yet. As of today, typically, if a web server supports QUIC, it conveys this information in the initial `HTTP HEADERS` reply sent back to the browser. This includes an alternative-service (ALT-SVC) field containing an Application-Layer Protocol Negotiation (ALPN) identifier. ALPN describes alternative HTTP versions available (e.g., `http/1.1`, `h2`, `h3`). If QUIC is supported, this ALPN identifier is set to `h3`. Upon detecting `h3` in the ALPN identifier, the browser establishes a fresh connection to the web server via QUIC.

Accordingly, an important point to note is that connection to a QUIC-enabled website generally involves both TCP and QUIC traffic. Furthermore, due to various other factors, e.g., resources hosted on third-party websites that may not support QUIC, even more TCP traffic may be generated.

B. Identifying QUIC-enabled websites

For data collection, we select Tranco's list of the top 1 million domains (https://tranco-list.eu/), often used for research purposes. As such lists do not provide any information regarding whether a website supports QUIC, our initial task is to create a list of top domains that are QUIC-enabled. For

this purpose, we leverage cURL and establish connections to each domain in Tranco's list, restricting the requests to `HTTP HEAD` only (instead of the whole index page), minimizing both client and server-generated traffic. The ALT-SVC field in the received `HTTP HEADERS` surely indicates whether the website supports QUIC. Consequently, we identified over 174,000 websites (out of the top 1 million from Tranco's list) that support QUIC.

The list we generate does not mandate that the servers contacted during the actual data collection phase will consistently support QUIC. This could be due to multiple factors, e.g., backend server configurations. Therefore, we carry out one more processing step. We revisit the aforementioned 174,000+ websites ten consecutive times. If QUIC support is found in all ten visits, we classify the website as QUIC-enabled.

C. Enabling DoQ for DNS resolutions

Contemporary web browsers (e.g., Firefox, Chrome, Safari) offer native support for DNS-over-HTTPS (DoH). However, if other forms of DNS resolution are needed (be it plain-text or encrypted), the browser needs to delegate to the underlying OS. For DoQ, we have only the second option, i.e., to configure the OS to use DoQ instead of the default plain-text DNS. For this purpose, we utilize AdGuard's DoQ-proxy, which intercepts plain-text DNS communications initiated by applications on the host system (e.g., the web browser) and sends them through encrypted QUIC transport channels to remote DoQ resolvers. This exclusive use of DoQ makes the attack easier to execute—both in terms of capturing packets and being budget-friendly regarding packet processing and storage costs—thereby making the attack more realistic to execute. Currently, the number of public DNS resolvers providing DoQ access is limited. Consequently, to fully comply with the proxy application, we rely on AdGuard's DoQ resolver, the first public DNS resolver to support the DoQ protocol.

D. Data collection process

1) Vantage points: We utilized remote clusters provided by CloudLab (https://cloudlab.us/), a well-established research infrastructure. Data was collected from four distinct vantage points (consisting of x86 servers running stock Ubuntu 22.04), namely, Utah, Massachusetts, Clemson, and Wisconsin, spanning across days to enhance the generalization of the models developed and to minimize encountering CAPTCHA pages.

2) Implementation: We developed a Docker container bundled with several Python and BASH scripts, facilitating the automation of the entire process. Within this container, we use the Selenium API to direct a Google Chrome browser to visit the QUIC-enabled domains identified. Each website visit is assigned a predefined timeout for deterministic completion; those that fail to load within the timeout are removed from the dataset.

For each scenario (cf. Sec. III-A), the containers visit each domain a specified number of times (see details below) to ensure a consistent amount of traces per domain for the training process. Concurrently, a separate container runs the DoQ-proxy for domain name resolution (cf. Sec. IV-C). `tcpdump` runs in the background to capture all packets traversing through the website visiting and the DoQ-proxy container's networking interfaces. For each website visit, we restart the DoQ-proxy connection and the browser to flush the caches (i.e., DNS caches and browser cache). The stored packet traces are subsequently processed using `tshark` to extract the important features and store them in CSV format for further data processing.

By executing these processes within distinct containers, we ensure adequate isolation to capture only the traffic relevant to web browsing. Accordingly, the network traffic corresponding to the top 500 QUIC-enabled websites, each visited 1,280 times, forms the dataset of the monitored websites. For the unmonitored websites, the network traffic data are due to visits (4 times each) to the top 74,700 QUIC-enabled websites from the range [500, 174,000+] ordered as per Tranco's ranking.

BENCHMARK DATASET.

- The dataset comprises network traffic generated from real browsing sessions using the emerging QUIC transport protocol, both for DNS and HTTP/3.
- Traffic corresponding to more than 75,000 websites is collected for closed and open world settings.
- Data was collected from multiple vantage points.
- To support reproducibility and future research, a containerized traffic collector and the dataset are released at https://github.com/cslev/DoQ_QUIC_webtraffic_analysis.

V. MODELING WEB TRAFFIC

The transformer model, known for its effectiveness in capturing long-range dependencies in sequence data with its self-attention mechanism [5], offers promise for website fingerprinting tasks. By treating network packets as words/tokens and utilizing self-attention mechanisms, it can effectively learn intricate patterns in packet sequences. Transformer models can be categorized into three main architectures: encoder-only, encoder-decoder, and decoder-only. For our work, we use the encoder-only architecture (see Fig. 2), which is commonly employed for classification tasks, including traffic modeling [13].

The encoder-only model we use consists of three core components: input embedding, encoder, and classification head. The input embedding layer maps the raw input features representing network traffic into high-dimensional vector representations. These embedded representations are then combined with positional encodings and a special `CLS` token. The encoder stack, which forms the core of the model, applies multi-headed self-attention mechanisms across the input packet sequence. The multi-head self-attention allows the model to weigh (i.e., learn the importance) of different packets and their features as needed; e.g., whether the first/last few DoQ packets are more important or not will be learned by the model heads independently. This allows the model to capture long-range dependencies and extract relevant features. The whole packet sequence is then represented concisely in the fixed-size `CLS` token. The classification head takes the encoded representation in the `CLS` token and passes it through dense layers, ultimately

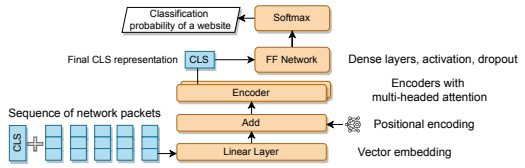


Fig. 2. Transformer model architecture for website fingerprinting

outputting the probability distribution over the target classes. The predicted class is then determined by selecting the class with the highest probability score.

Based on the transformer architecture, we develop two models for the two scenarios we are interested in—$\mathcal{S}^{\text{DoQ}}$ and $\mathcal{S}^{\text{DoQ+H3}}$. First, we explain how network traffic is represented.

A. Input traffic representation

Each visit to a website results in packet communications on the wire. We sort the packets of a website visit as per their timestamps and thereby obtain a time-ordered sequence of packets in both directions. Recall, the traffic generated during a website visit are DoQ packets for domain resolutions, and QUIC and TCP packets for browsing contents (cf. Sec. IV-A).

As we utilize sequence-based transformer models for WFP, a data point for modeling is a sequence of the first k packets representing a single trace of a website visit, where each packet is represented by a fixed number of features (defined below).

B. Scenario $\mathcal{S}^{DoQ}$: Modeling DoQ Traffic

We first describe how we build a WFP model, $\mathcal{T}^{\text{DoQ}}$, for the scenario $\mathcal{S}^{\text{DoQ}}$ (cf. Sec. III-A).

Features play a crucial role in modeling data and influence model performance. When traffic is encrypted, as is the case with DoQ traffic, we are limited to extracting meta-information. For each packet, we extract direction (i.e., incoming/outgoing), inter-arrival time (IAT), and packet size (in bytes). Note that IAT is the time between the current packet and the previous packet (irrespective of the direction). Thus, we represent each packet with a concise 3-element vector.

As mentioned above, a data point for modeling is a sequence of k packets of a trace, where in the case of DoQ, each packet is represented by a 3-dimensional vector. For traces with fewer than k packets, the remaining part of the input is padded with a special vector (e.g., $[-255, -255, -255]$ to fill the sequence length to k, to inform the model of non-existent packets in the sequence. Conversely, for traces having more than k packets, the excess packets are truncated. Following this pre-processing step, normalization is applied to refine the quality of the trace representation. The parameter k is crucial: if set too low, the model may not capture enough packets and fail to learn effective fingerprinting. If k is too high, excessive padding can occur, which, based on our observations, results in poorer performance and longer computational times.

Hence, the value of k should be such that it captures most of the DoQ packets in a trace in $\mathcal{S}^{\text{DoQ}}$. We observed that 95% of the DoQ traces in our dataset have a DoQ packet count of less than 150. If we set $k = 200$, then we capture all DoQ packets in almost all traces. In light of this observation, we pragmatically set our maximum sequence length k to 200 for effective DoQ traffic representation and model training.

We use $\mathcal{T}^{\text{DoQ}}$ to refer to the model trained on the packet sequences from DoQ traffic generated during website visits.

C. Scenario $\mathcal{S}^{DoQ+H3}$: Modeling DoQ+QUIC Traffic

With this scenario, we analyze if adding QUIC web traffic would improve performance of $\mathcal{S}^{\text{DoQ}}$. Therefore, in addition to the features defined for DoQ traffic, we use two more features. Namely, we record if the transport protocol is QUIC or TCP along with another boolean value indicating if a packet is part of a DoQ (domain resolution) or HTTP (browsing) traffic. Using these five features, we train a transformer model on packet sequences from traffic related to website visits and refer to the model as $\mathcal{T}^{\text{DoQ+H3}}$.

VI. Performance Evaluations

We compare the two transformer models, $\mathcal{T}^{\text{DoQ}}$ and $\mathcal{T}^{\text{DoQ+H3}}$, corresponding to the two scenarios, $\mathcal{S}^{\text{DoQ}}$ and $\mathcal{S}^{\text{DoQ+H3}}$, respectively. We compare these transformer models with the previous best-known sequence model, LSTM (Long Short-Term Memory) that has been proposed in the literature for website fingerprinting [14]. We apply the LSTM model for the scenario $\mathcal{S}^{\text{DoQ+H3}}$, giving it packet sequences with the same features as for $\mathcal{T}^{\text{DoQ+H3}}$ (cf. Sec. V-C). The LSTM architecture consists of four hidden layers, followed by a dense layer and a `softmax` for classification. We evaluate the WFP models in both closed world and open world settings.

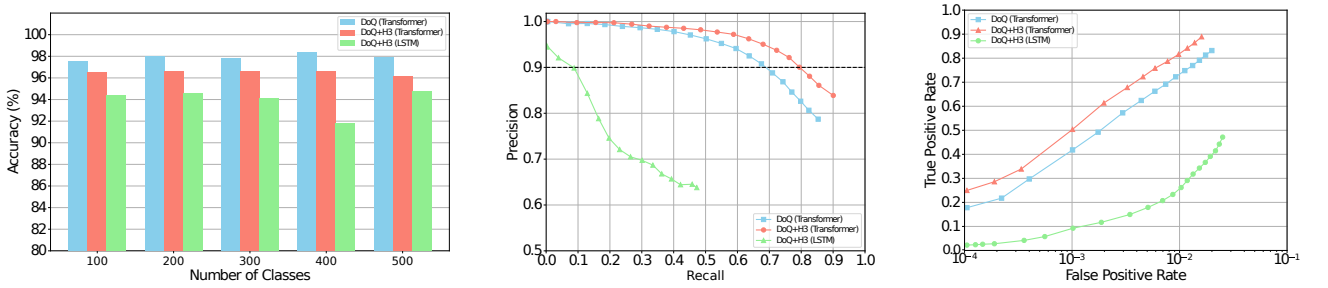
A. Evaluations in Closed World

In the closed world, the attacker is assumed to know all the websites the victim visits, called the monitored websites. The attacker's goal is to identify which monitored website is visited. This is a multi-class classification problem, where each website visit must be classified into one of the monitored websites. As a result, all three website fingerprinting models are trained as multi-class classifiers. For example, in an experiment with 500 websites, the models are trained to classify a website visit into one of these 500 websites.

To evaluate the models, we utilize the complete dataset consisting of 1280 traces per monitored website (500 x 1,280). We adopt an 80:20 train-test split, with 1024 traces per website used for training and the remaining 256 traces used for testing Recall that the training and testing traces come from all four locations. (cf. Sec. IV-D1).

1) Metrics: WFP attack in the closed world is evaluated using *accuracy*, defined as the ratio of the correctly identified monitored websites to the total number of monitored websites.

2) Hyper-parameter tuning: We first carry out experiments to set the hyper-parameters for the transformer models. We focus on the three important parameters: embedding size, number of attention heads, and the number of encoders. For this purpose, we choose $\mathcal{T}^{\text{DoQ+H3}}$ as the model, as it is trained with more features and data (DoQ, QUIC, and TCP flows). The number of websites ranges from 100 to 500.



(a) Closed world performance; accuracy vs. number of websites (classes).

(b) Open world performance: precision vs. recall. 90% precision is emphasized for easy comparison.

(c) Open world performance: TPR at low FPR values. Note, X-axis is in log scale.

Fig. 3. Performance of the models in the closed and open world settings.

We first set the number of encoders to one and varied the embedding size and the number of heads. Note that embedding size increases along with the number of attention heads since the number of attention heads must be a factor in embedding size in the model architecture. By increasing the values of embedding size and attention heads (as tuples), we found that *embedding size of 32 and 16 attention heads* gives the highest accuracy; hence we select and fix these values. Next, we vary the number of encoders from one to four. We observed that the performance does not improve beyond two encoders, and therefore we set the *number of encoders to two* for our models. We note that the transformer models are relatively small with less than 750K trainable parameters.

3) Results: We evaluate the accuracy of the three models, namely LSTM, $\mathcal{T}^{\text{DoQ}}$, and $\mathcal{T}^{\text{DoQ+H3}}$, in the closed world. Fig. 3a plots the results. All three models achieve high accuracy in this setting. Both transformer models show higher accuracy than the baseline LSTM model. Between the two transformer models, it is interesting to observe that $\mathcal{T}^{\text{DoQ}}$ outperforms $\mathcal{T}^{\text{DoQ+H3}}$, although the difference is not significant. Yet, to understand better, we analyze the traces in detail.

Recall our discussion on sequence length (k) selection in Sec. V-B—almost all DoQ traces in $\mathcal{S}^{\text{DoQ}}$ scenario (i.e., DoQ-only traces) have less than 200 packets. Therefore, $k = 200$ includes most DoQ packets in $\mathcal{S}^{\text{DoQ}}$ scenario. But in $\mathcal{S}^{\text{DoQ+H3}}$ scenario, there are HTTP packets besides the DoQ packets. Therefore, we extracted the index of the last DoQ packet in website traces in the $\mathcal{S}^{\text{DoQ+H3}}$ scenario where DoQ packets are shuffled with HTTP packets (e.g., when multiple domains are being resolved). We found that the index of the last DoQ packet in the traces, on average, is around 6000; hence the same sequence length would (ideally) be required. Such extremely long sequences create multiple challenges, importantly, high computational time for training, large storage space, and a high cost of buffering packets in a network middlebox. Nevertheless, we evaluated the classification performance of $\mathcal{T}^{\text{DoQ+H3}}$ model for higher sequence lengths from 500 to 1000 and noticed only marginal gains ($< 1\%$) in accuracy. This indicates that when DoQ is mixed with other traffic, increasing DoQ packets (beyond a limit) has diminishing returns. Therefore, for the rest of the experiments, we maintain the same sequence length, i.e., $k = 200$.

B. Evaluations in Open World

1) Website fingerprinting models: We now consider the open world setting, wherein the attacker maintains a set of monitored websites, and any website not on this list is considered to be unmonitored. For our experiments here, we set the number of monitored websites to the top-100 QUIC-enabled domains, each with 360 traces (i.e., 36,000 in total). We train the models with 101 classes, where the additional class is for the unmonitored traces. The unmonitored class consists of the top 45,000 websites (after the top-100) with 4 traces per website, totaling 180,000 traces. Note, for the unmonitored websites, there is no overlap between training and testing sets; i.e., a website seen in the training set is not present in the testing set. Also, the traces are randomly picked from different locations for both training and testing (cf. Sec. IV-D1). The train-test partition for monitored classes is 75:25, and for the unmonitored class is 1:1. This leaves 27,000 (9000) monitored traces and 90,000 (90,000) unmonitored traces for training (testing); thus, we have a high 1:10 ratio between monitored to unmonitored traces for testing.

Evaluation strategy: During inference, given a trace to classify, the model gives the probability of it being of any of these 101 classes. For evaluations, we follow a common approach; we consider this prediction probability to be the confidence the model has in its prediction. Therefore, if the maximum probability given by the classifier is less than a pre-defined confidence threshold, we reclassify it to the unmonitored class. This aims for a lower false positive rate (FPR). Since the number of unmonitored websites can be one or more orders of magnitude higher than the monitored websites, even an FPR of 10^{-1} is considered high for practical attack utility. Therefore, we focus on evaluating the models at FPRs of 10^{-2} and lower. The attacker can select the confidence threshold such that the classifier produces a low FPR even if it misses identifying some monitored websites (false negatives).

2) Metrics: To evaluate the open world scenario, we use the traditional metrics of True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN), wherefrom Precision, Recall (i.e., TPR), and FPR are derived.

3) Results: Fig. 3b plots the precision-recall curves for the three models in the open world experiments. The transformer models clearly outperform the LSTM model. At a high 90% precision, the recall achieved is less than 10% with the LSTM model for the $\mathcal{S}^{\text{DoQ+H3}}$ scenario. Whereas, at the same 90%

precision, the $\mathcal{T}^{\text{DoQ+H3}}$ model achieves a significantly high recall of $\approx 80\%$. Although the $\mathcal{T}^{\text{DoQ}}$ model edged out $\mathcal{T}^{\text{DoQ+H3}}$ in the closed world experiments, in the more challenging open world experiments, $\mathcal{T}^{\text{DoQ+H3}}$ achieves 10% more recall at 90% precision in comparison to $\mathcal{T}^{\text{DoQ}}$, with the latter achieving (still) a high 70% recall. Note that, by modeling only DoQ traffic, $\mathcal{T}^{\text{DoQ}}$ still has 60% higher recall than the LSTM model, although the LSTM model utilizes both DoQ and HTTP traffic.

Fig. 3c plots TPR (recall) at low FPR values. We have similar observations as before. At 10^{-2} FPR, the LSTM model achieves only $\approx 25\%$ TPR, whereas the transformer models achieve much higher TPR values. Specifically, $\mathcal{T}^{\text{DoQ}}$ has a TPR of $\approx 73\%$, whereas the TPR of $\mathcal{T}^{\text{DoQ+H3}}$ is greater than 80% at 10^{-2} FPR. At an even lower FPR of 10^{-3}, the TPR of LSTM drops to a very low $\approx 10\%$, whereas the $\mathcal{T}^{\text{DoQ+H3}}$ model is still able to identify 50% of the monitored websites.

TAKEAWAYS.
i) Just by modeling DoQ traffic, an attacker with a modest traffic capturing budget (in terms of processing and storing network packets) is able to identify 70% of monitored websites (recall) at 90% precision.
ii) Modeling both DoQ and web traffic, an attacker achieves an even higher recall of $\approx 80\%$ at 90% precision.
iii) Transformer models are superior to the LSTM model in fingerprinting QUIC-enabled website traffic.

VII. CONCLUSIONS

We conducted the first comprehensive study exposing the vulnerability of the latest protocols—DoQ for DNS resolutions and QUIC for web—against WFP attacks targeting user privacy. This work opens up further research directions, e.g., would a more realistic scenario involving multi-tab browsing pose challenges for WFP? As packet padding is known to be ineffective [15], a future direction is to explore application-level defenses.

Acknowledgment: This research/project is supported by the National Research Foundation, Singapore, and the Cyber Security Agency of Singapore under the National Cybersecurity R&D Programme and the CyberSG R&D Programme Office (Award CRPO-GC2-ASTAR-001). Any opinions, findings, conclusions, or recommendations expressed in these materials are those of the author(s) and do not reflect the views of the National Research Foundation, Singapore, the Cyber Security Agency of Singapore, or the CyberSG R&D Programme Office.

REFERENCES

[1] A. Panchenko, F. Lanze, J. Pennekamp, T. Engel, A. Zinnen, M. Henze, and K. Wehrle, "Website Fingerprinting at Internet Scale," in *NDSS*, 2016.
[2] A. Master and C. Garman, "A worldwide view of nation-state internet censorship," *Free and Open Communications on the Internet*, 2023.
[3] R. Marx, "Head-of-Line Blocking in QUIC and HTTP/3: The Details," Blogpost, https://calendar.perfplanet.com/2020/head-of-line-blocking-in-quic-and-http-3-the-details/, Dec 2020 [Accessed: Nov 2024].
[4] M. Kosek, L. Schumann, R. Marx, T. V. Doan, and V. Bajpai, "DNS privacy with speed? Evaluating DNS over QUIC and its impact on web performance," in *ACM IMC '22*, 2022.
[5] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *NIPS*, 2017.
[6] S. E. Oh, S. Sunkam, and N. Hopper, "p^1-FP: Extraction, Classification, and Prediction of Website Fingerprints with Deep Learning," *PoPETS*, 2019.
[7] J. Yan and J. Kaur, "Feature Selection for Website Fingerprinting," *PoPETS*, 2018.
[8] G. Cherubin, R. Jansen, and C. Troncoso, "Online Website Fingerprinting: Evaluating Website Fingerprinting Attacks on Tor in the Real World," in *31st USENIX Security Symposium*, Aug. 2022, pp. 753–770.
[9] J.-P. Smith, P. Mittal, and A. Perrig, "Website Fingerprinting in the Age of QUIC," *PoPETS*, 2021.
[10] S. Siby, M. Juárez, C. Díaz, N. Vallina-Rodriguez, and C. Troncoso, "Encrypted DNS -> Privacy? A Traffic Analysis Perspective," in *NDSS Symposium*, 2020.
[11] L. Csikor, H. Singh, M. S. Kang, and D. M. Divakaran, "Privacy of DNS-over-HTTPS: Requiem for a Dream?" in *IEEE Euro S&P*, 2021.
[12] G. Huang, C. Ma, M. Ding, Y. Qian, C. Ge, L. Fang, and Z. Liu, "Efficient and low overhead website fingerprinting attacks and defenses based on tcp/ip traffic," in *ACM Web Conf.*, 2023, pp. 1991–1999.
[13] B. Wu, P. Gysel, D. M. Divakaran, and M. Gurusamy, "ZEST: Attention-based Zero-Shot Learning for Unseen IoT Device Classification," in *IEEE NOMS*, 2024, pp. 1–9.
[14] V. Rimmer, D. Preuveneers, M. Juárez, T. van Goethem, and W. Joosen, "Automated Website Fingerprinting Through Deep Learning," in *Proc. NDSS*, 2018.
[15] S. D. Siby, L. Barman, C. A. Wood, M. M. Fayed, N. Sullivan, and C. Troncoso, "Evaluating practical QUIC website fingerprinting defenses for the masses," *PoPETS*, 2023.

Levente Csikor (cslev@cslev.vip) is a research scientist at the Institute for Infocomm Research (I^2R), A*STAR, Singapore. His research explores the intricate dimensions of security and privacy in next-generation networks.

Lian Ziyue (daryllzy@gmail.com) is a software engineer at China Merchants Bank, Xiamen, China. His research interests include the application of AI, natural language processing (NLP), and recommender systems.

Haoran Zhang (haoran.zhang@u.nus.edu) obtained his Master's degree from the National University of Singapore.

Nitya Lakshmana (nityalak@comp.nus.edu.sg) is a Lecturer at the School of Computing, National University of Singapore. Her research interests include networking and network security, with a focus on 4G/5G security.

Dinil Mon Divakaran (Senior Member, IEEE; dinil_divakaran@i2r.a-star.edu.sg) is a Senior Principal Scientist at the Institute for Infocomm Research (I^2R), A*STAR, Singapore. He is also an Adjunct Assistant Professor at the School of Computing at the National University of Singapore. His research interests are network security, web security, and AI security.