Domain Name Encryption Is Not Enough: Privacy Leakage via IP-based Website Fingerprinting

Preprint · February 2021

CITATIONS
0

READS
100

4 authors, including:

Nguyen Phong Hoang
Stony Brook University
25 PUBLICATIONS **190** CITATIONS
SEE PROFILE

Arian Akhavan Niaki
University of Massachusetts Amherst
14 PUBLICATIONS **125** CITATIONS
SEE PROFILE

Nguyen Phong Hoang, Arian Akhavan Niaki, Phillipa Gill, and Michalis Polychronakis

Domain Name Encryption Is Not Enough: Privacy Leakage via IP-based Website Fingerprinting

Abstract: Although the security benefits of domain name encryption technologies such as DNS over TLS (DoT), DNS over HTTPS (DoH), and Encrypted Client Hello (ECH) are clear, their positive impact on user privacy is weakened by—the still exposed—IP address information. However, content delivery networks, DNS-based load balancing, co-hosting of different websites on the same server, and IP address churn, all contribute towards making domain–IP mappings unstable, and prevent straightforward IP-based browsing tracking.

In this paper, we show that this instability is not a roadblock (assuming a universal DoT/DoH and ECH deployment), by introducing an IP-based website fingerprinting technique that allows a network-level observer to identify *at scale* the website a user visits. Our technique exploits the complex structure of most websites, which load resources from several domains besides their primary one. Using the generated fingerprints of more than 200K websites studied, we could successfully identify 84% of them when observing solely destination IP addresses. The accuracy rate increases to 92% for popular websites, and 95% for popular *and* sensitive websites. We also evaluated the robustness of the generated fingerprints over time, and demonstrate that they are still effective at successfully identifying about 70% of the tested websites after two months. We conclude by discussing strategies for website owners and hosting providers towards hindering IP-based website fingerprinting and maximizing the privacy benefits offered by DoT/DoH and ECH.

Keywords: Domain Name Encryption, DoT, DoH, Encrypted Client Hello, Website Fingerprinting

DOI 10.2478/popets-2021-0058
Received 2021-02-28; revised 2021-06-15; accepted 2021-06-16.

Nguyen Phong Hoang: Stony Brook University, E-mail: nghoang@cs.stonybrook.edu
Arian Akhavan Niaki: University of Massachusetts - Amherst, E-mail: arian@cs.umass.edu
Phillipa Gill: University of Massachusetts - Amherst, E-mail: phillipa@cs.umass.edu
Michalis Polychronakis: Stony Brook University, E-mail: mikepo@cs.stonybrook.edu

1 Introduction

Due to the increase of Internet surveillance in recent years [13, 31], users have become more concerned about their online activities being monitored, leading to the development of privacy-enhancing technologies. While various mechanisms can be used depending on the desired level of privacy [48], encryption is often an indispensable component of most privacy-enhancing technologies. This has led to increasing amounts of Internet traffic being encrypted [3].

Having a dominant role on the Internet, the web ecosystem thus has witnessed a drastic growth in HTTP traffic being transferred over TLS [28]. Although HTTPS significantly improves the confidentiality of web traffic, it cannot fully protect user privacy on its own when it comes to preventing a user's visited websites from being monitored by a network-level observer. Specifically, under current web browsing standards, the domain name information of a visited website can still be observed through DNS queries/responses, as well as the Server Name Indication (SNI) field of the TLS handshake packets. To address this problem, several domain name encryption technologies have been proposed recently to prevent the exposure of domain names, including DNS over TLS (DoT) [51], DNS over HTTPS (DoH) [49], and Encrypted Client Hello (ECH) [88].

Assuming an idealistic future in which all network traffic is encrypted and domain name information is never exposed on the wire as plaintext, packet metadata (e.g., time, size) and destination IP addresses are the only remaining information related to a visited website that can be seen by a network-level observer. As a result, tracking a user's browsing history requires the observer to infer which website is hosted on a given destination IP address. This task is straightforward when an IP address hosts only one domain, but becomes more challenging when an IP address hosts multiple domains. Indeed, recent studies have shown an increasing trend of websites being co-located on the same hosting server(s) [47, 95]. Domains are also often hosted on multiple IP addresses, while the dynamics of domain–IP mappings may also change over time due to network configuration changes or DNS-based load balancing.

Given these uncertainties in reliably mapping domains to their IPs, we investigate the extent to which accurate browsing tracking can still be performed by network-level adversaries based solely on destination IPs. In this work, we introduce a *lightweight* website fingerprinting (WF) technique that allows a network-level observer to identify with high accuracy the websites a user visits based solely on IP address information, *enabling network-level browsing tracking at scale* [74]. For instance, an adversary can use—already collected by existing routers—IPFIX (Internet Protocol Flow Information Export) [101] or NetFlow [2] records to easily obtain the destination IP addresses contacted by certain users, and track their browsing history.

For our attack, we first crawl a set of 220K websites, comprising popular and sensitive websites selected from two website ranking lists (§5). After visiting each website, we extract the queried domains to construct a domain-based fingerprint. The corresponding IP-based fingerprint is then obtained by continuously resolving the domains into their IPs via active DNS measurement (§4.1). By matching these IPs from the generated fingerprints to the IP sequence observed from the network traffic when browsing the targeted websites, we could successfully fingerprint 84% of them (§6.3). The successful identification rate increases to 92% for popular websites, and 95% for popular *and* sensitive websites.

To further enhance the discriminatory capacity of the fingerprints, we consider the critical rendering path [38] to capture the approximate ordering structure of the domains that are contacted at different stages while a website is being rendered in the browser (§4.2). Our results show that the enhanced fingerprints could allow for *91% of the tested websites* to be successfully identified based solely on their destination IPs(§6.4).

Given the high variability of website content and domain–IP mappings across time, we expect that once generated, a fingerprint's quality will deteriorate quickly over time. To assess the aging behavior of the fingerprints, we conducted a longitudinal study over a period of two months. As expected, fingerprints become less accurate over time, but surprisingly, after two months, they are still effective at successfully identifying about 70% of the tested websites (§7).

As our WF technique is based on the observation of the IPs of network connections that fetch HTTP resources, it is necessary to evaluate the impact of HTTP caching on the accuracy of the fingerprints. This is because cached resources can be loaded directly from the browser's cache when visiting the same website for a second time, resulting in the observation of fewer connections per fingerprint. Furthermore, our attack exploits the fact that websites often load many external resources, including third-party analytics scripts, images, and advertisements, making their fingerprints more distinguishable. We thus also investigated whether the removal of these resources due to browser caching or ad blocking could help to make websites less prone to IP-based fingerprinting (§8).

By analyzing the HTTP response header of the websites studied, we find that 86.1% of web resources are cacheable, causing fewer network connections to be observable by the adversary if these resources are loaded from the browser's cache (§8.1). Moreover, using the Brave browser to crawl the same set of websites, we found that the removal of third-party analytics scripts and advertisements can impact the order in which web resources are loaded (§8.2), significantly reducing the accuracy of the enhanced fingerprints from 91% to 76%. Nonetheless, employing the initially proposed WF technique in which the critical rendering path [38] is not taken into account, we could still fingerprint 80% of the websites even when browser caching and ad blocking are considered.

Regardless of the high degree of website co-location and the dynamics of domain–IP mappings, our findings show that domain name encryption alone is not enough to protect user privacy when it comes to IP-based WF. As a step towards mitigating this situation, we discuss potential strategies for both website owners and hosting providers towards hindering IP-based WF and maximizing the privacy benefits offered by domain name encryption. To the extent possible, website owners who wish to make IP-based website fingerprinting harder should try to (1) minimize the number of references to resources that are not served by the primary domain of a website, and (2) refrain from hosting their websites on static IPs that do not serve any other websites. Hosting providers can also help by (1) increasing the number of co-located websites per hosting IP, and (2) frequently changing the mapping between domain names and their hosting IPs, to further obscure domain–IP mappings, thus hindering IP-based WF attacks.

2 Background and Motivation

In this section, we review some background information on domain name encryption technologies and discuss the motivation behind our study. In particular, we highlight how our IP-based fingerprinting attack is different from prior works, allowing network-level adversaries to effectively mount the attack at scale.

2.1 Domain Name Encryption

In today's web browsing environment, there are two channels through which domain name information is exposed on the wire: plaintext DNS requests/responses, and the Server Name Indication (SNI) extension of TLS.

The plaintext nature of DNS not only jeopardizes user privacy, but also allows network-level entities to interfere with user connections. For example, an on-path attacker can inject forged DNS responses to redirect a targeted user to malicious hosts [27]. The domain name information exposed via DNS packets and the SNI field has also been intensively exploited by state-level network operators for censorship purposes [14, 46, 75, 78, 82]. To cope with these security and privacy problems, several solutions have been recently proposed to safeguard domain name information on the wire, including DoT [51], DoH [49], and ECH [88].

By encrypting DNS traffic, DoT/DoH preserves the integrity and confidentiality of DNS resolutions. Several companies (e.g., Google [35], Cloudflare [4]) offer free DoT/DoH services to the public, while popular web browsers including Chrome and Firefox already support DoH [35, 67], with the latter enabling DoH (through Cloudflare) by default for users in the US since 2019. However, the design choice of these vendors to centralize all DNS resolutions to one trusted resolver has raised several privacy concerns. As a result, more privacy-centric DNS resolution schemes have also been proposed, including DoH over Tor [71, 91], Oblivious DNS [92, 97], and distributed DoH resolution [44], to not only conceal DNS packets from on-path observers, but also to deal with "nosy" recursors.

SNI has been incorporated into the TLS protocol since 2003 [17] as a workaround for name-based virtual hosting servers to co-host many websites that support HTTPS. During the TLS handshake, the client includes the domain name of the intended website in the SNI field in order for the server to respond with the corresponding TLS certificate of that domain name. Until TLS 1.2, this step takes place before the actual encryption begins, leaving the SNI field transmitted in plaintext, and exposing users to similar security and privacy risks as discussed above. TLS 1.3 provides an option to encrypt the SNI field, concealing the domain name information [87]. Since March 2020, ESNI has been reworked into the ECH extension [88]. In order for ECH to function, a symmetric encryption key derived from the server's public key has to be obtained in advance. This public key can be obtained via an HTTPS resource record lookup. Thus, it is important to note that ECH cannot provide any meaningful privacy benefit without the use of DoT/DoH, and vice versa. Mozilla has supported ECH since Firefox 85 [52].

2.2 Website Fingerprinting

Website fingerprinting (WF) is a type of traffic analysis attack, employed to construct fingerprints for a set of websites based on the traffic pattern observed while browsing them. Depending on which metadata is visible from the encrypted traffic, different WF techniques can be used to determine whether a user under surveillance visited any of the monitored websites.

Numerous WF attacks targeting anonymized or obfuscated communication channels have been proposed [40, 58, 73, 79, 85, 98, 107, 108], in which the actual destination IP address is hidden by means of privacy-enhancing network relays [32, 48], such as Tor [26] or the Invisible Internet Project (I2P) [43, 113]. However, WF attacks on standard encrypted web traffic (i.e., HTTPS), in which no privacy-enhancing network relays are employed, have not been comprehensively investigated, especially at the IP-address level. This is because the domain name information previously available in several plaintext protocols (e.g., DNS, the SNI extension of TLS, and OCSP queries [90]) can be easily obtained from the network traffic (§2.1). This information alone can already be used for a straightforward inference of applications or web services being visited [42, 102]. However, in an idealistic future where domain name encryption (i.e., DoT/DoH *and* ECH) is fully deployed, visibility to any information above the IP layer will be lost. Under these conditions, and given the high degree of web co-location [47], our ultimate goal is to investigate the extent to which websites can still be fingerprinted at scale, based solely on the IP address information of the servers being contacted.

Given the numerous WF methods introduced in the past, one may wonder *why do we even need another website fingerprinting method?* In addition to the aforementioned reasons and pitfalls of previous WF techniques [53], the rationale behind our fingerprinting technique based solely on IP-level information stems from the increasing deployment of domain name encryption technologies [25, 61]. Currently, most web traffic does expose domain information, as domain name encryption has not been fully deployed [103], and thus network-level browsing tracking at scale through DNS or SNI is much easier. However, once this massive-scale monitoring capability is gone due to DoT/DoH and ECH, the next

best option for ISPs to continue tracking at a similar scale will be to rely on IP addresses, which are already collected as part of IPFIX (Internet Protocol Flow Information Export) [101] or NetFlow [2] records by existing routers. Although more elaborate fingerprinting schemes can certainly be conceived for HTTPS traffic, these will require a significant deployment effort and cost [74], both for constructing and maintaining the fingerprints, as well as for matching them.

3 Threat Model

Internet service providers (ISPs) have been increasingly harvesting user traffic for monetization purposes, such as targeted advertising [18, 34, 55]. Our threat model considers the real-world scenario in which a local adversary (e.g., an ISP) passively monitors users' traffic and attempts to determine whether a particular website was visited. The adversary carries out the following steps to create website fingerprints.

First, the adversary visits a set of websites and records all domain names that are contacted to fetch their resources. A domain-based fingerprint for each website is then built from this set of contacted domains. After that, these domains are periodically resolved to their hosting IPs, which are used to construct IP-based fingerprints. Depending on the relationship between a domain name and its hosting IP(s), a connection to a unique IP can be used to easily reveal which website is being visited if the IP only hosts that particular domain name. Finally, to conduct the WF attack, the adversary tries to match the sequence of IP addresses found in the network trace of the monitored user with the IP-based fingerprints constructed in the previous step to infer which website was visited.

The effectiveness of our attack depends on two primary factors, namely, the *uniqueness* (§6) and *stability* (§7) of the fingerprints. It is worth emphasizing that our model does not assume fingerprinting of obfuscated network traffic, in which the IP address information is already hidden by means of privacy-enhancing technologies. This class of attacks, whose goal is to use sophisticated traffic analysis methods to fingerprint anonymized network traffic, has been extensively investigated by prior studies [20, 40, 58, 79, 98]. Our threat model requires only minimal information collected from the network traffic, i.e., destination IPs.

We consider a browsing scenario in which one website is visited at a time, which is particularly valid when it comes to ordinary web users on devices with smaller screens, and is also most often the case of casual browsing behavior (except, perhaps, the rare event of a browser restart with many previously open tabs). Although some users may visit more than one website at a time, they mostly interact with one tab at a time. There is also a time gap when changing from one tab to another to open or reload a different website, especially for users with a single screen. All these together allow an observer to distinguish between individual website visits, as also evident by existing techniques that can be employed to split a network trace of such multi-tab activity into individual traces [23, 24, 111]. Moreover, although many individual users may be located behind the same NAT network, Verde et al. [105] have developed a framework that can identify different individuals behind a large metropolitan WiFi network based on NetFlow records. To keep our study simple, we thus assume that the adversary already employs the aforementioned techniques to obtain the network trace of different individuals before conducting our WF attack.

4 Fingerprint Construction

Next, we explain how we construct IP-based fingerprints in more detail, from creating the initial domain-based fingerprints to deriving the final IP-based fingerprints. At a conceptual level, we first explore the straightforward approach of resolving all domains loaded while visiting a website to their hosting IPs, from which we create a set of unique IPs that can potentially be used as the IP-based fingerprint for that website. We then take the critical rendering path [38] into account to improve the fidelity of the fingerprints, by considering the approximate order in which domains are loaded while the website is being rendered on the screen.

4.1 Basic IP-based Fingerprint

Assuming an idealistic web browsing scenario in which domain name information can no longer be extracted from network traffic due to the full deployment of domain name encryption, the only remaining information visible to the adversary is packet metadata (e.g., time, size) and sequences of connections to remote IP addresses of contacted web servers. Under these conditions, the adversary would need to fingerprint targeted websites based primarily on this information. As introduced in our threat model, the adversary first visits the targeted websites and records all domains that are contacted while browsing each website. Each domain can then be resolved to its hosting IP address(es). As a result, the mapping between domains and hosting IP ad-

dress(es) is the basic unit on which the adversary relies to construct IP-based fingerprints.

When browsing a website, the browser first contacts the web server to fetch the initial resource—usually an HTML document. It then parses the HTML document and subsequently fetches other web resources referenced. Based on this underlying mechanism of fetching a webpage, the adversary constructs domain-based fingerprints as follows. For a given website, its domain-based fingerprint consists of two parts: the *primary* domain, denoted as d_p, and a set of *secondary* domains, denoted as d_s. The fingerprint then can be represented as: $d_p + \{d_{s_1}, d_{s_2}, ..., d_{s_n}\}$. The primary domain is the domain of the URL shown in the browser's address bar, and typically corresponds to the server to which the first connection is made for fetching the initial HTML document of the visited webpage. Secondary domains may be different from the primary one and are used for hosting other resources needed to load the webpage.

From the domain-based fingerprints constructed above, the adversary can then obtain their corresponding IP-based fingerprints by repeatedly resolving the domain names into their hosting IP(s). Given that a domain name may be resolved to more than one IP, each domain in a domain-based fingerprint is converted to a set of IP(s) with at least one IP in it. As domain-based fingerprints are comprised of two parts, the inherent structure of IP-based fingerprints also consists of two parts. The first part contains the IP(s) of the primary domain name, while the second part is a set of sets of IPs obtained by resolving the secondary domains. The fingerprint then can be represented as:

$$\{d_p ip_1, d_p ip_2, ..., d_p ip_n\} + \{\{d_{s_1} ip_1, d_{s_1} ip_2, ..., d_{s_1} ip_n\},$$
$$\{d_{s_2} ip_1, d_{s_2} ip_2, ..., d_{s_2} ip_n\}, ..., \{d_{s_n} ip_1, d_{s_n} ip_2, ..., d_{s_n} ip_n\}\}$$

To simplify the construction and matching of fingerprints, we reduce the above fingerprint by considering the union of the sets of IP addresses of all secondary domains (second part of the above fingerprint). The simplified version of the IP-based fingerprint can thus be represented by just two sets of IP addresses as:

$$\{d_p ip_1, d_p ip_2, ..., d_p ip_n\} + \{d_{s_1} ip_1, d_{s_1} ip_2, ..., d_{s_1} ip_n,$$
$$d_{s_2} ip_1, d_{s_2} ip_2, ..., d_{s_2} ip_n, ..., d_{s_n} ip_1, d_{s_n} ip_2, ..., d_{s_n} ip_n\}$$

Although it might seem that this simplification discards some part of the structural information of the page, we found no significant difference in accuracy when evaluating both fingerprint formats. Therefore, we opt to use the latter fingerprint structure, as it is simpler and allows for faster matching.

4.2 Enhanced IP-based Fingerprint with Connection Bucketing

When a webpage is visited, besides the initial connection to the primary domain, multiple requests may then be issued *in parallel* to fetch other resources referenced in the initial HTML document. Once fetched, these resources may sometimes trigger even more requests for other sub-resources (e.g., JavaScript). The absolute order of these requests on the wire can change from time to time, depending on many uncertain factors, such as the performance of the upstream network provider and the underlying operating system. For that reason, we did not consider the order in which domains are contacted when constructing the domain-based fingerprints, and thus we gather all secondary domains into one bucket, as described in §4.1. However, when viewing all these requests as a whole, there still exists a high-level ordering relationship that we can capture when considering the critical rendering path [38]. Specifically, there are certain render-blocking and critical objects that always need to be loaded prior to some other objects.

The chain of events from fetching an initial HTML file to rendering the website on screen is referred to as the *critical rendering path* [38]. When visiting a website, the browser first contacts the primary domain to fetch the initial HTML file (e.g., `index.html`). Next, this file is parsed to construct the DOM (Document Object Model) tree. The browser then fetches several web resources from remote destinations to render the webpage. Depending on the complexity of the webpage, these resources may include HTML, JavaScript, CSS, image files, and third-party resources, which may in turn load more sub-resources hosted on other third-party domains [76]. According to the Internet Archive, a typical website loads an average of 70 web resources as of this writing [7]. When considering this critical rendering path, there are three important events that we can use to cluster connections into three "buckets:" *domLoading*, *domContentLoaded*, and *domComplete*.

domLoading is triggered when the browser has received the initial HTML file and parses it to construct the DOM tree. As a result, multiple parallel connections to fetch critical resources referenced by the DOM tree are initiated right after this event is fired.

domContentLoaded is triggered when both the DOM and CSSOM (CSS Object Model) are ready [38], signaling the browser to create the render tree. The event is typically fired without waiting for style sheets, images, and subframes to load [69]. After this event, subsequent connections can often be observed for fetching

elements such as non-blocking style sheets, JavaScript files, images, and subframes.

domComplete is triggered when the website and its sub-resources have been loaded. After this event is fired, non-essential objects can still be downloaded in the background, leaving the critical rendering path unaffected. For example, external JavaScript files are known to be render-blocking and are recommended to be moved to the end of the webpage or to be included with a `defer` attribute of the `<script>` tag [36]. Therefore, a small cluster of connections can often be observed after this event is triggered.

Based on these observations about the critical rendering path, we enhance the structure of our fingerprints (both domain-based and the corresponding IP-based ones) to comprise the primary domain and three sets of domains corresponding to the three events aforementioned. The enhanced domain-based fingerprint can then be represented as:

$$d_p + \{d_{s_1}, d_{s_2}, ...\} + \{d_{s_2}, d_{s_3}, ...\} + \{..., d_{s_{n-1}}, d_{s_n}\}$$

Note that a domain can appear in more than one bucket if multiple objects are fetched from that domain at different times. Accordingly, the representation of the enhanced IP-based fingerprint follows the same structure, comprising four sets of IP addresses: i) the set of IP addresses of the primary domain, and ii) three sets of IP addresses corresponding to the three buckets above.

5 Experiment Setup

In this section, we provide the details of how we set up and conducted our experiments for assessing the effectiveness of IP-based website fingerprinting. In particular, we discuss the rationale behind our test list of websites, and the duration and location of our measurement.

5.1 Selection of Test Domains

From an adversarial point of view, it is desirable for an attacker to be able to reveal as many websites as possible. It is, however, impractical to crawl the entire Internet, given that there are more than 362.3 million domain names registered across all top-level domains (TLDs) as of 2020 [8]. In addition, many of them are dormant or even unwanted domains [100] that the majority of Internet users will never visit. As our goal is to assess the extent to which domain name encryption would prevent the leakage of the majority of users' browsing activities via IP-based website fingerprinting, we opt to focus on those websites that are legitimately visited in real-world scenarios. Therefore, we choose to use domains from the Tranco top-site ranking list [104], since it has been shown to have a good overlap with the web traffic observed by the Chrome User Experience Report [84].

The research community often relies on one of the four top-site lists (Alexa [9], Majestic [11], Umbrella [12], and Quantcast [5]). However, studies have discovered several issues with these lists that can negatively impact research outcomes if not handled properly [89, 104]. To remedy the shortcomings of these lists, the Tranco list is curated to aggregate the four aforementioned lists, resulting in a list of more than seven million domains. Even then, due to the dynamic nature of the web [15, 57], there are still domains that are unstable, not responsive, or do not serve any web content in the Tranco list, especially in its long tail [84, 89]. Therefore, we select the top 100K popular domains from the Tranco list for our study, because any ranking under 100K is not statistically significant, as suggested by both top-list providers and previous studies [9, 89].

While some websites are so common that visiting them may be considered to be a very low privacy risk (e.g., `facebook.com`, `twitter.com`, or `youtube.com`), the leakage of visits to more "sensitive" websites is definitely an important concern. This is even more so in oppressive regions, where browsing certain online content could be considered as a violation of local regulations [39, 72, 112]. To that end, we complement our dataset by manually choosing websites from the Alexa list that belong to categories deemed "sensitive,"[1] such as LGBT, sexuality, gambling, medical, and religion. In total, our dataset consists of 220,743 domains, including the top 100K popular domains of the Tranco list[2] and 126,597 domains from Alexa's sensitive categories. Among these, there are 5,854 common domains between the two data sources.

Although one may consider our test list as a closed-world dataset, it is infeasible to repeatedly crawl the entire Internet, which has more than 362.3M domains registered at the time of our experiment [8]. It is also unlikely that a network-level adversary is interested in

1 It is worth noting that sensitivity can be different from site to site, depending on who, when, and from where is visiting the site [42]. We intuitively choose these complementary domains based on our common sense of what is sensitive based on those categories that are often blocked by many Internet censors around the world [75].

2 The list was created on March 3rd 2020, and is available at https://tranco-list.eu/list/J2KY.

fingerprinting all websites on the Internet. However, the ability of continuously conducting our WF attack on more than 220K domains highlights the scalability of our method. In fact, we cover an order of magnitude more domains compared to previous WF attacks against DoT/DoH traffic [19, 50, 96], in which the largest open-world setting comprised fewer than 10K domains [19].

5.2 Measurement Duration and Location

Prior work often overlooks the temporal aspect when constructing fingerprints. More specifically, many efforts are conducted in a one-off manner or over a short period of time, neglecting the temporal characteristics of fingerprints whose websites' content may evolve over time [57]. For our work, in which we focus on IP-based fingerprints, the churn in domain–IP mappings is also another major concern [45]. Due to the variability of website content and hosting IPs across time, a previously constructed fingerprint may not be valid after a certain time period. Therefore, a longitudinal measurement study is essential to examine the robustness of fingerprints, which in turn impacts the efficacy of their use in WF attacks.

Over a period of 60 days (from March 5th to May 3rd, 2020), we repeatedly crawled the 220K websites from our test list curated in §5.1, using the Chrome browser (desktop version 80.0), running on Ubuntu 20.04 LTS. When visiting each website, we extract all domains contacted to construct the fingerprint for that website using the steps discussed in §4. At the network level, we capture the sequence of destination IPs contacted, to evaluate the accuracy of our IP-based WF method (§6). Note that the collection of this sequence of IPs is oblivious to the domains extracted independently when loading each website.

Due to DNS-based load balancing, many domains, especially of popular websites, may map to different IP addresses at different times [45]. Therefore, once the set of domains that were contacted to render each website is extracted, we continuously resolve them to obtain their IP addresses until the next crawl. This best-effort approach allows us to obtain as many IPs as possible for those domains that employ DNS-based load balancing. However, to make sure our experiment does not saturate DNS servers (thus affecting other legitimate users), we enforce a rate limit of at least three hours. In other words, contacted domains of a website are only resolved again if they were not resolved within the last three hours. The entire process for each crawl batch takes approximately 2.5 days. As a result, we have collected a total of 24 data batches during a two-month period. To stimulate future studies in this research domain, we make our dataset available to the research community at `https://homepage.np-tokumei.net/publication/publication_2021_popets`.

Our measurement is conducted from a cluster of machines located in a gigabit academic network in the US. Due to the rapid increase in the use of content delivery networks (CDN), web content can be served from multiple servers distributed across different locations, depending on the origin of the request [45]. Although our dataset can be considered as representative for web users within our geographical area, it would have missed some IP addresses of CDN-hosted websites which can only be observed at other locations. Nonetheless, as mentioned in §3, the adversary in our threat model is local and also has access to the Internet from the same network location as the monitored users (e.g., the ISP of a home or corporate user), and will not observe any other IPs of CDN-hosted websites either. Adversaries at different locations can always set up machines within their network of interest and conduct the same experiment with ours to construct a dataset of fingerprints that matches those websites browsed by users within the network of their control.

6 Fingerprinting Accuracy

Next, we evaluate the accuracy of our WF techniques using the data collected in §5. We begin with an analysis of the information entropy that we can expect from domain-based fingerprints, and then evaluate the accuracy of IP-based fingerprints.

6.1 Fingerprint Entropy

As discussed in §3, the creation of IP-based fingerprints is based on the domains contacted while visiting the targeted websites. Therefore, it is important to first examine the uniqueness of these domains, as it impacts the effectiveness of IP-based WF. This will aid us in deciding whether a domain should be included or not as part of a fingerprint. For example, if a certain domain is contacted when visiting every single website, then there is no point in including it. In contrast, if a unique domain is only contacted when visiting a particular website, it will make the fingerprint more distinguishable. The more unique a domain is, the higher the information entropy that can be gained [94], resulting in a better fingerprint.

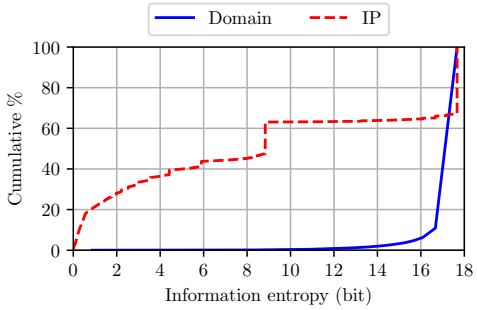

Fig. 1. CDF of the information entropy gained per domain/IP as a percentage of all the unique domains/IP addresses observed.

Let $P(d)$ be the probability that a particular domain will be contacted when visiting a given website among the targeted websites. The information entropy in bits gained if that particular domain is contacted can be calculated using the following formula:

$$\text{Information entropy} = -\log_2 P(d)$$

The blue (solid) line in Figure 1 shows the entropy gained from each domain as a percentage of the approximately 475K unique domains observed in each crawl batch. Almost 90% of these domains are unique to the website from which they are referenced, yielding high information entropy (17.7 bits). In contrast, there are a few domains from which we can only gain a small amount of entropy. This result aligns well with the study of Greschbach et al. [37] in which the Alexa top-site list was crawled and for 96.8% of the websites there exists at least one domain that is unique only to these websites.

Table 1 lists the top-ten domains that provide the least information entropy. Eight of them belong to Google and two belong to Facebook. These domains are commonly included in many websites, thus only contributing a small amount of entropy. For instance, `www.google-analytics.com` is included in more than half of the websites. However, we opt to keep them as part of our fingerprints, as most of them still provide more than one bit of information, helping to differentiate between websites that reference these external Google/Facebook services and those that do not.

Based on the entropy for each domain computed above, we then calculate the entropy gained per IP address, since our ultimate goal is to perform WF at the IP level. Given an observed IP, there are two possibilities regarding the domain(s) it may correspond to. First, the IP may be associated with only a single domain, in which case its entropy can be deduced directly from the domain's entropy. Second, the IP may co-host multiple domains. In this case, the IP's entropy is calculated by taking the average of the entropy values of all domains (that have been observed to be) hosted on it. Note that

Table 1. Top-ten domains that yield the lowest entropy.

Domain name	# Websites	Entropy
`www.google-analytics.com`	114K (55%)	0.87
`fonts.gstatic.com`	102K (49%)	1.03
`fonts.googleapis.com`	102K (49%)	1.04
`www.google.com`	76K (37%)	1.44
`stats.g.doubleclick.net`	72K (35%)	1.53
`www.googletagmanager.com`	64K (31%)	1.71
`www.facebook.com`	53K (25%)	1.97
`connect.facebook.net`	53K (25%)	1.98
`googleads.g.doubleclick.net`	49K (24%)	2.09
`ajax.googleapis.com`	34K (16%)	2.62

calculating the entropy using both average and median gives us similar results because most co-hosted domains on the same IP addresses often provide a similar amount of information entropy. We thus choose the former one.

Considering these two possibilities, we then calculate the information entropy of the 340K IPs observed from our continuous DNS measurement in each crawl batch. As indicated by the red (dashed) line of Figure 1, 50% of the IPs provide at least 9 bits of information entropy, while there is a group of more than 30% of the IPs that provide a high amount of information entropy (17.7 bits), which correspond to IPs hosting only a single domain.

6.2 Primary Domain to IP Matching

Before evaluating our WF techniques, we first investigate whether WF is needed at all. Prior studies have shown that a significant fraction of websites have a one-to-one mapping between primary domains and their hosting IP(s) [45]. This first connection can be distinguished from the subsequent requests since there is usually a noticeable time gap during which the browser needs to contact the primary domain to download the initial HTML file, parses it, and constructs the DOM tree before multiple subsequent connections are initiated to fetch referenced resources. As a result, it is straightforward for an adversary to target this very first connection to infer which website is being visited.

More specifically, when a domain is hosted on one IP or multiple IPs without sharing its hosting server(s) with any other domains, it is easy to infer the domain from the IP(s) of its hosting server(s). We analyzed the DNS records of all primary domains in our dataset to quantify the fraction of websites that can be fingerprinted by just targeting their primary IP(s). We find that 52% of the websites studied have their primary domain hosted on their own IP(s), while the remaining

Table 2. Percentage of successfully identified websites using i) naive primary domain to IP matching (§6.2), ii) basic fingerprinting (§4.1, §6.3), and iii) enhanced fingerprinting with connection bucketing (§4.2, §6.4).

Website type	Total	Primary Domain	IP-based Fingerprinting	Connection Bucketing
All websites crawled	208,191	107,455 (52%)	174,662 (84%)	**189,527 (91%)**
Popular websites	93,661	58,989 (63%)	86,147 (92%)	90,231 (96%)
Sensitive websites	120,293	51,538 (43%)	93,988 (78%)	104,983 (87%)
Sensitive and popular	5,763	3,072 (53%)	5,473 (95%)	5,687 (99%)

48% are co-hosted on a server with at least another website. This result means that an adversary can already infer 52% of the targeted websites based solely on the IP address of the very first connection to the primary domain, without having to consider secondary connections. The third column of Table 2 shows the breakdown of these websites in terms of their popularity and sensitivity. Note that the total number of websites shown here is lower than the total number of test websites selected in §5 due to some unresponsive websites when conducting our experiment.

6.3 Basic IP-based Website Fingerprinting

To fingerprint the remaining 48% of websites whose primary domains are co-hosted, an adversary would need to analyze the second part of their IP-based fingerprint that captures the IP addresses of secondary domains.

Going back to the way we build our IP-based fingerprints in §4.1, the basic IP-based fingerprint has two parts. The first part consists of the primary domain's IP(s), and the second part comprises a set of IPs obtained by resolving all secondary domains. Given a sequence of IPs $[ip_0, ip_1, ip_2, ..., ip_n]$ observed from a network trace, we first scan ip_0 against the primary part of all IP-based fingerprints, which are created by repeated active DNS measurements (§5.2). If ip_0 is found among the primary IPs of a given fingerprint, the fingerprint is added to a pool of candidates. We then compare the *subset* $\{ip_1, ip_2, ..., ip_n\}$ with the secondary part of each candidate fingerprint. For each matching IP, we add the entropy provided by that IP to the total amount of entropy gained for that particular candidate fingerprint. Finally, we choose the fingerprint with the highest total entropy to predict the website visited.

Using this IP-based WF method we obtained an increased matching rate of 84%—that is, 84% of the websites in our data set were identified with 100% accuracy. The breakdown of the fingerprinted websites is shown in the fourth column of Table 2. Among these fingerprinted websites, we could precisely match 92% of the popular websites and 78% of the sensitive websites. More worrisome is the fact that 95% of sensitive *and* popular websites can be fingerprinted.

6.4 Enhanced Website Fingerprinting with Connection Bucketing

We next evaluate the effectiveness of the enhanced WF (§4.2), in which we take the critical rendering path into consideration to cluster IPs into three buckets. Similarly to the basic fingerprints (§6.3), given a sequence of IPs $[ip_0, ip_1, ip_2, ..., ip_n]$, we first scan ip_0 against all fingerprints to create a pool of candidate fingerprints. For the subsequence $[ip_1, ip_2, ..., ip_n]$, our goal is to split it into three buckets of connections that can potentially be matched with the three buckets of IPs in the IP-based fingerprints. Based on the time of each connection initiation captured at the network level (§5.2), we use k-means clustering [30, 60, 63] to split them into three sets of IPs.

For every candidate fingerprint, we intersect each bucket of IPs in the fingerprint to the corresponding bucket of IPs captured from the network trace. Then, for each matching IP, we add its entropy to the total amount of entropy gained for that particular fingerprint. Finally, we choose the fingerprint with the highest entropy to predict the visited website.

Using this approach, the accuracy rate can be improved to 91%. The breakdown of fingerprinted websites is shown in the last column of Table 2. For the popular and the sensitive websites, we obtain an accuracy rate of 96% and 87%, respectively. However, a more alarming result is that 99% of sensitive *and* popular websites can be precisely fingerprinted, posing a severe privacy risk to their visitors.

We now look into the remaining 9% (18,664) of websites for which we could not find an exact match. As shown in Figure 2, 20% of these websites have only two matching candidates, while about 50% of them have only up to ten matching candidates. By manually examining some of these fingerprints, we found many cases in which the matching candidate domains actually point to the *same* website (but without redirect-

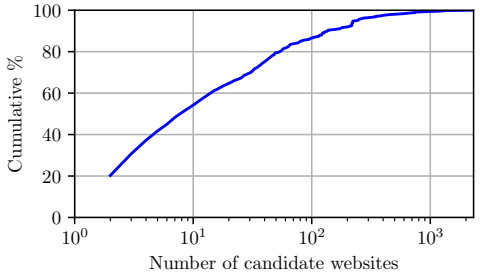

Fig. 2. CDF of candidate websites per fingerprint for the remaining 9% (18,664) of websites that could not be matched.

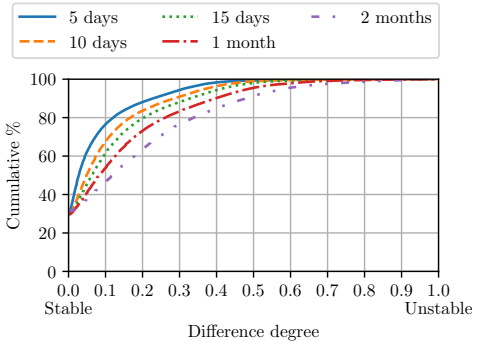

Fig. 3. CDF of the stability of domain names loaded in each website as a percentage of all websites studied.

ing to the same domain). These are mostly owned by organizations who have registered the same name under different TLDs (e.g., `bayer.de` vs. `bayer.com`), or variations of the domain (e.g., `christianrock.net` vs. `christianhardrock.net`) to protect their brand against domain squatting [100]. A more determined adversary could invest the effort to implement more advanced techniques for identifying such duplicate websites. For instance, string similarity can be used to cluster similar domains, while image similarity can be used to group websites with similar screenshots of the start page.

7 Fingerprint Stability

As mentioned in our threat model, the efficacy of a WF attack also depends on the stability of fingerprints over time. There are two primary reasons why a website fingerprint may go stale. First, the website may change over time with existing elements removed and new elements added [57]. Second, the mapping between a domain and its hosting IP(s) may also change [45]. Consequently, a previously constructed fingerprint may no longer be valid after a certain time period.

Since our IP-based fingerprints are constructed based on domains contacted while browsing the targeted websites, we first examine the extent to which these websites are stable in terms of the domains that they reference. We introduce a *difference* metric to quantify the change in this set of domains for a given website over time as follows. Let D_{t_0} and D_{t_1} be the sets of contacted domains observed when browsing a website at time t_0 and t_1, respectively, the *difference degree* for this website is calculated as:

$$\text{Difference degree} = \frac{(D_{t_0} \cup D_{t_1}) - (D_{t_0} \cap D_{t_1})}{D_{t_0} \cup D_{t_1}}$$

Based on this definition, we consider a website as *stable* during a period $t_0 \rightarrow t_1$ when the set of domains observed at time t_1 have not changed compared to those previously observed at time t_0, yielding a *difference degree* of 0. In contrast, a website is considered as *unstable* when its difference degree is 1, meaning that all domains observed at time t_1 are different from those previously seen at t_0.

Figure 3 shows the stability of the websites studied in terms of the domains contacted while visiting them. About 30% of the websites contact the exact same set of domains to download web resources for the whole two-month period of our study. Within a five-day period, 80% of the websites are still almost completely stable, with a difference degree lower than 0.1, while this percentage decreases to 50% over the two-month period. Understandably, almost half of the websites we study are the most popular on the Internet. Hence, it is expected that their content will be changed or updated on a regular basis. However, even after two months, almost 80% of the websites are still stable, with a difference degree lower than 0.3, meaning that 70% of observed domains are still being used to host web resources needed to render these websites.

This is a favorable result for the adversary, as it shows that domains are an effective and consistent feature. The result particularly implies that the adversary does not need to keep crawling all websites repeatedly to construct domain-based fingerprints. Based on the results of Figure 3, the adversary perhaps can divide websites into two groups comprising stable and less stable websites. For instance, the stable group consists of 80% of websites with a difference degree lower than 0.2 after ten days, while the less stable group consists of the remaining 20% of websites. Then, the adversary would only need to re-crawl the less stable ones every ten days to keep their domain-based fingerprints fresh, instead of all websites.

However, in our threat model, what can be actually observed by the adversary is only IP addresses. We thus apply the same difference formula to quantify the

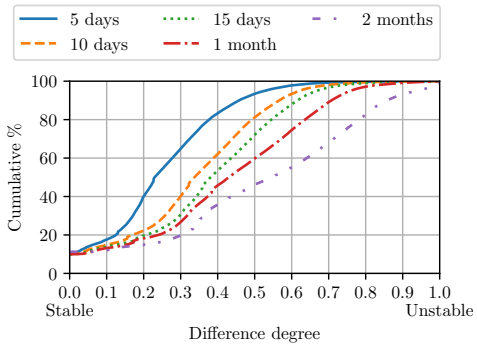

Fig. 4. CDF of the stability of IP addresses in each IP-based fingerprint as a percentage of all websites studied.

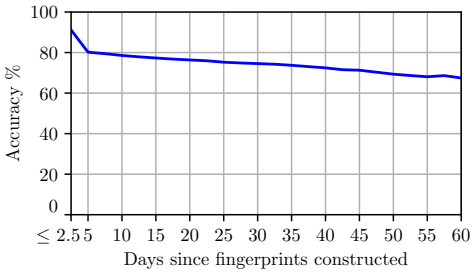

Fig. 5. Fingerprint stability over time. Even after two months, 70% of the tested websites can still be fingerprinted.

stability of IP-based fingerprints. Unlike domain-based fingerprints, IP-based fingerprints become stale faster, as shown in Figure 4. We find that only 10% of the IP-based fingerprints contain the same set of IPs over the course of two months. After ten days, 60% of the fingerprints have more than 30% of their IPs changed. After two months, half of the IPs have changed in more than 50% of the fingerprints.

Given these results, we investigate how the instability of the IP-based fingerprints impacts the accuracy of our WF attack. We consider an attack scenario in which the adversary uses fingerprints constructed in the past to track the users' browsing activities at a future time. Figure 5 shows the accuracy (i.e., the percentage of successfully identified websites) of our enhanced WF approach over the course of two months. Within 2.5 days[3] after their generation, our fingerprints consistently yield a high accuracy of 91%. Over the course of two months, we can see a gradual decrease in the accuracy. However, this decrease is quite modest, as after five to ten days since their construction the fingerprints can still be used to accurately identify about 80% of the websites. This number only decreases to about 70% after two months.

Although IP-based fingerprints go stale faster compared to their domain-based fingerprints, those IP addresses that change frequently mostly correspond to secondary domains, and only a small fraction corresponds to primary domains (see Appendix A for details). The vast majority of primary domains are hosted on mostly static IP addresses for the whole period of our study. As a result, the persistently stable IP addresses of these primary domains in the IP-based fingerprints is the reason why our IP-based fingerprints are still effective at revealing 70% of the targeted websites even though a large number of IP-based fingerprints have changed significantly after two months, as indicated in Figure 4.

The above finding means that the adversary can intelligently split domains into two groups, based on previously observed data. The first group consists of domains whose IPs are dynamic, while the second group contains domains whose IPs remain static over a configurable amount of time. The adversary then only needs to periodically perform DNS lookups for the first group after a desired amount of time has passed, depending on the network overhead and resources the adversary can sustain for conducting the attack.

8 Fingerprint Robustness

We next examine the impact of HTTP caching on the effectiveness of our WF since resources are often cached by web browsers to improve websites' performance. In addition, our WF also exploits the fact that websites often load external resources, including images, style sheets, fonts, and even "unwanted" third-party analytics scripts, advertisements, and trackers [76], which result in a sequence of connections to several servers with different IPs, making the fingerprints more unique. Thus, we also investigate whether blocking these unnecessary resources would help make websites less distinguishable, thus reducing their fingerprintability.

8.1 Impact of HTTP Caching on Website Fingerprinting Accuracy

When a website is revisited, cached resources can be served from the local cache without the browser fetching them again from their origins. Since our attack is based solely on the observation of the IP of connections to remote destinations, we are interested in examining the fraction of cacheable resources and the extent to which HTTP caching impacts the effectiveness of our WF.

3 The lowest time granularity is 2.5 days because each crawl batch in our dataset requires this amount of time to be collected, as discussed in §5.

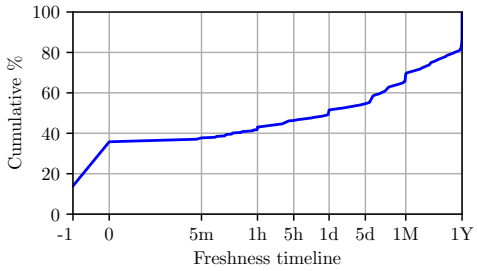

Fig. 6. CDF of the freshness timeline of HTTP resources. (m: minute, h: hour, d: day, M: month, Y: year)

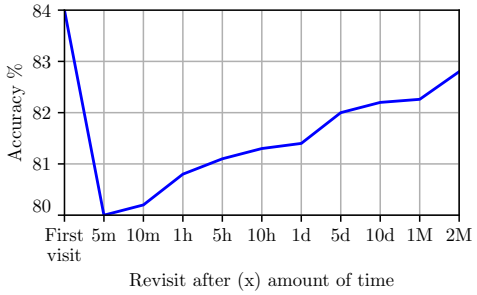

Fig. 7. The accuracy of IP-based fingerprinting attack when excluding destinations where cacheable resources are hosted. (m: minute, h: hour, d: day, M: month)

Analyzing the response header of 21.3M objects observed while crawling the tested websites, we find that 86.1% of them are cacheable. In other words, these HTTP resources can be stored and served from the local cache without being downloaded again from the remote servers when being revisited.

Utilizing the cache-control information in the HTTP response header, we compute the freshness timeline for each resource. The freshness timeline is the amount of time during which the browser can store and serve resources from its cache without downloading them again from their original servers. Figure 6 shows the distribution of the freshness timeline of 21.3M objects. The value "-1" denotes uncachable resources (13.9%) that must be downloaded again from their origin if revisited, while "0" indicates cacheable resources (21.9%) that always need to be revalidated with their origin. In other words, these two types of resources will always cause a network connection to their original servers if revisited. On the other hand, the remaining 64.2% of resources can be loaded directly from the local cache without making any network connections.

Next, we evaluate the impact of cacheable resources on our attack accuracy by excluding IPs on which cacheable resources are hosted. We use the basic fingerprinting method here for our evaluation (§4.1) instead of the enhanced one (§4.2), because revisited resources may not be freshly loaded in the order of the critical rendering path as in the first visit.

Although many web resources are cached, we could still obtain a high accuracy. As shown in Figure 7, even when websites are revisited after only five minutes, meaning that the majority of resources can be served from the local cache, an accuracy of 80% can still be obtained—a decrease of just 4% (from 84%) compared to when websites are visited for the first time. If websites are revisited after one hour, one day, or one month, our basic WF attack can obtain a gradually increased accuracy of 80.8%, 81.4%, and 82.3%, respectively.

There are two primary reasons why our attack is still highly effective although the majority of resources are cacheable. First, excluded IPs often host long-term cacheable shared resources, such as fonts and JavaScript code, which contribute only a small amount of entropy to the fingerprint if included. Second, for cacheable resources hosted on IPs with high entropy, not all resources have the same freshness timeline. In fact, we find that half of the origins that host resources cacheable for more than one hour also serve another resource with a freshness timeline shorter than five minutes, causing at least one network connection to the original server if revisited.

8.2 Fingerprinting Under Ad Blocking

Due to the prevalence of ads and analytics scripts that harvest users' information [86], many advertisement and tracker blocking tools have been developed to protect user privacy. Of these tools, the Brave browser has stood out to be one of the best browsers for user privacy on the Clearnet to date [56].

Therefore, we opt to use the Brave browser for investigating the impact of ad blocking on IP-based WF. During the last four batches of our data collection process (i.e., ten days), at the same time with crawling the test websites (without blocking ads and trackers), we instrumented the Brave browser (desktop version 1.6.30) to crawl these websites a second time. While the Brave browser is loading each website, we also capture (1) the set of domains contacted to fetch web resources for rendering the website, and (2) the sequence of IPs contacted to fetch web resources.

Using the same fingerprinting techniques as in §6, we then match the sequences of IP addresses observed while browsing with the Brave browser to our IP-based fingerprints. As expected, the fingerprinting accuracy

rate decreases from 91% to 76% when using the enhanced fingerprints. Since our enhanced fingerprinting approach (§4.2) primarily relies on the ordering structure in which web resources are loaded, the partial removal of these resources by the Brave browser has impacted the effectiveness of this approach.

However, when employing the initial (basic) fingerprinting approach (§4.1), in which the ordering structure of loaded resources is not considered, we could still obtain an accuracy rate of 80%. This is due to the removal of external resources (e.g., analytics scripts, tracking images) whose information entropy is not significant. We provide a further analysis of the filtered domain names in Appendix B.

In practice, the use of ad blocking (if any) by a client may not be explicit to an adversary since IPs are the only information that can be observed. To detect the use of ad blocking at the client side, the adversary can obtain up-to-date IP blocklists of ads/trackers from well-maintained sources (e.g., FireHOL [10]) to examine if there are connections to IPs of servers where those ads/trackers are hosted. Then, the adversary can decide which matching mode to employ for obtaining a higher fingerprinting accuracy. Note that the data collection procedure does not change regardless of the mode.

9 Countermeasures

We next discuss potential directions for website owners and hosting providers toward making IP-based WF more challenging, thus maximizing the privacy benefits provided by domain name encryption.

9.1 Website Owners

Our WF exploits the fact that websites typically load resources from multiple servers. From the viewpoint of a network observer, this makes their fingerprints more distinguishable. External resources such as ads and tracking scripts served from third-party domains may sometimes fetch even more "unwanted" objects from other third-party domains without the knowledge of the website owner [76]. As shown in §8, blocking these objects hinders (to some extent) the fingerprintability of a website. Owners who wish to provide increased privacy to their visitors can thus minimize the inclusion of third-party resources. On the other hand, privacy-conscious users can use ad and tracker blocking tools to make their browsing activities harder to fingerprint.

Another reason for contacting domains that are different from the primary domain is the web design strategy known as *domain sharding* [68]. Since traditional web browsers limit the number of concurrent connections per remote server according to the HTTP/1.1 specification [29], website owners often host resources on different domains as a workaround to improve the page load time by parallelizing connections to multiple servers. However, the introduction of HTTP/2 makes this strategy irrelevant.

Among the many new features of HTTP/2, *server push* and *request multiplexing* play an important role in improving page load time [110]. By eliminating round-trip requests, the server can preemptively push referenced resources to reduce latency. With multiplexing, resource requests can be sent in parallel through a single TCP connection. To gain any performance benefits offered by these new mechanisms, it is recommended to co-host web resources on the same server [110]. From the perspective of IP-based fingerprinting, this is a welcome change that will aid in reducing the fingerprintability of websites, as a network-level observer will now see only one connection stream to a single remote IP address.

9.2 Hosting providers

Even with HTTP/2 and all resources served from the same domain, if a website is exclusively hosted alone on the same IP(s), it can still be trivially fingerprinted. Hosting providers can aid in hindering IP-based WF by maximizing two factors: the co-location degree of websites and the dynamics of domain–IP mappings.

We have shown that websites that are not co-hosted with other websites are the most vulnerable to our attack due to the one-to-one mapping between their domain and hosting IP address(es). When a website is co-hosted with many other websites, it becomes more challenging to fingerprint—assuming their owners have taken the steps discussed in §9.1. Otherwise, despite a relatively high level of co-location of more than 1K websites, we could still successfully fingerprint them because their fingerprints are unique enough to differentiate a given website from the rest of the co-hosted websites, as shown in Appendix C.

In addition to increasing the co-location degree, hosting providers can also maximize the dynamics of domain–IP mappings to hinder WF further. By analyzing the dynamics of mappings between domains and IPs throughout the whole period of our study, we find that it is feasible to increase the dynamics of domain–IP mappings from the perspective of hosting providers. However, we only observe this behavior for a small number of primary and secondary domains, whereas almost

80% of the studied websites have their primary domains hosted on static IPs, allowing network-level observers to easily fingerprint them (see Appendix A for details).

10 Limitations

In this section, we discuss the limitations and the rationale behind the experimental design of our study, especially in relation to the critical evaluation conducted by Juarez et al. [53].

Most prior WF studies are often criticized for only considering a very small number of websites in a closed-world setting [53, 83]. However, it would be infeasible to crawl the entire Internet of more than 362.3 million domains registered across all TLDs [8], many of which are dormant domains that most users would never visit [100]. We thus use the Tranco top-site ranking list [104] to focus on those websites that are likely visited by most legitimate Internet users in real-world scenarios [84]. Moreover, our test list is not only curated from the most 100K popular domains, but is also complemented by more than 100K less popular but sensitive domains §5.1.

We believe that this is a reasonable trade-off for the breadth of coverage, which yields a manageable yet representative set of test domains, allowing us to conduct our experiment in a longitudinal fashion to shed light on the aging behavior of fingerprints (§7)—an important factor that is often not considered by prior studies. Regardless of the test list size, our WF attack was conducted in a closed-world setting. In a truly open-world setting, as the number of websites increases, the proposed WF technique may become more error-prone [53, 80, 83].

Another criticized assumption often made by previous studies is that the adversary can collect data under the same conditions (e.g., network connection, web browser, website localization) as the victim [53]. This is a valid criticism, especially when it comes to WF attacks on Tor traffic, because visiting the same domain via different Tor paths (exit nodes) may result in different localized versions of the website. The adversary in our threat model, however, is a local attacker (e.g., ISPs, corporate network administrators) who is in the same network with the victim. Therefore, it is straightforward for the adversary to set up an environment that is similar to that of the victim. Specifically, the availability of several OS and device fingerprinting tools based on the different implementations of the TCP/IP stack [16, 93], together with well-known "home-phoning" traffic of different web browsers [56], can assist the adversary in filtering background noise and resembling a similar browsing environment with the victim.

To keep our experiment manageable, we opt to use the Chrome browser for data collection because it is the most popular at the time of this study, occupying about 65% of the browser market share [6]. Although the cross-browser fingerprinting result in §8.2 has showed that our basic WF technique can still achieve an accuracy rate of 80%, we acknowledge that this accuracy could be impacted in more complex scenarios if different extensions and preferences are configured in the browser.

Our dataset is created by visiting the start page of the test websites without going into any subpages or interacting with them. We thus may have missed some characteristics of individual pages that could only be captured if some user interaction was involved (e.g., logging in). However, similarly to DNS-based monitoring, our attack model does not aim to distinguish between different links, pages, or events under the same website, which has been studied previously [64, 70]. Instead, the primary goal of the proposed WF technique is to determine if a given website was visited.

When conducting our attack, the resources of each website are considered independently for that given website while there could be cases in which more than one website is visited from the same browser, resulting in same resources (e.g., font, CSS, or JavaScript files) being shared among the websites. We have shown in the analysis of our fingerprint robustness (§8) that excluding these resources from the fingerprints due to browser caching or ad blocking can significantly impact the effectiveness of our enhanced WF technique (§4.2). Nonetheless, their removal does not completely thwart our attack, as we could still identify 80% of the websites studied using the basic WF technique (§4.1). This is because these shared resources are often hosted on common IP addresses that contribute only a small amount of entropy to the fingerprints when included.

Finally, the proposed privacy-enhancing countermeasure of increasing website co-location can lead to another privacy concern related to hosting centralization [59]. While this is a valid concern, this suggestion is (1) for hosting providers who are already chosen by the website owners to host their websites, and (2) based on the already centralized nature of the web, which has been an increasing trend for the last decade [47, 95]. Note that the adversary in our threat model (§3) corresponds to local attackers, such as ISPs and corporate network administrators, but not hosting providers or website owners. If a user's privacy goal is to conceal their online activities from hosting providers and web

owners, browsing via privacy-enhancing network relays (e.g., Tor [26]) would be a more suitable option.

11 Related Work

Encrypted network traffic analysis has gained more attention in recent years as the Internet is on its way to be fully encrypted since obtaining a TLS certificate has become free of charge and easier than ever [3]. However, some initial concept of this class of attack has been established since the 90s [106]. As one of the first attempts to apply traffic analysis on WF, Andrew Hintz simply counts the number of downloaded files and their size based on the number of connections, generating fingerprints for a set of targeted websites [41]. Similarly, Sun et al. [99] conduct a large-scale study on the fingerprintability of 100K webpages based on the number of objects requested as part of each website's download. However, this attack vector is no longer effective due to the introduction of persistent HTTP (default since HTTP/1.1 [29]) in which multiple files can be transmitted over a single TCP connection.

In the same year of the two studies above, the pre-alpha version of Tor was released [1], bringing online privacy to another level by not only encrypting the network traffic but also hiding the fact that a Tor user is browsing a particular website from both local network observers and the remote web server [26]. Since then, the literature has witnessed numerous studies on WF attacks on Tor using various techniques, ranging from classical machine learning methods [40, 58, 79, 108] to advanced deep neural networks [73, 98].

Similar to any other privacy-enhancing communications (e.g., Tor), encrypted DNS traffic is susceptible to traffic analysis. Therefore, padding was added to remedy this problem [66]. However, recent studies find that current padding strategies are not sufficient to cope with traffic analysis. Bushart et al. [19] show that padded encrypted DNS traffic is still vulnerable to traffic analysis attacks. Based on the size and timing information of encrypted DNS packets, the authors could deanonymize 86.1% of 10K websites studied. Using the sequence of bytes as a key feature to build a model for classifying encrypted DoH traffic, Siby et al. [96] could obtain a precision of 94% on a dataset of 5K domains. In another related work, Houser et al. [50] analyze DoT traffic using a classifier based on numerous statistical features extracted from the time of DNS packets, obtaining an accuracy of 83% for a dataset of 98 websites. Compared to the scale of our measurement, these prior studies employ several machine learning techniques on much smaller datasets, with the largest open-world dataset comprising only 10K domain names [19].

When WF attacks are designed based primarily on traffic features, such as packet size and burst, they can be thwarted by obfuscating or adding noise to the traffic, as is evident by a series of defensive techniques for Tor proposed previously [20, 22, 33, 54, 62, 77, 109]. Siby et al. [96] has indeed come to a conclusion that routing DoH traffic via Tor can effectively mitigate their WF attack. There have been several implementations of DoH over Tor [71, 91], which can help to remedy the situation. This is the fundamental reason why we care about fingerprinting at the IP level, and refrain from using other traffic features. Specifically, while DoT/DoH traffic can be obfuscated by tunneling via Tor to cope with these prior attacks, our attack does not target the DoT/DoH traffic itself but the actual destination IPs contacted when a website is visited, which are more challenging to hide or obfuscate. One may suggest the use of Tor in this case as a countermeasure. Nonetheless, it is important to stress that the fundamental privacy risk that domain encryption techniques aim to address is orthogonal to those of Tor.

In terms of attacks using the IP address information, Hoang et al. [45] assess the privacy benefits offered by domain name encryption by simply resolving domains into IPs and estimate their co-location degree without actually visiting any websites. The authors conclude that co-hosting can help to improve privacy. While this observation is valid, our WF method could still achieve a high accuracy rate regardless of many co-hosted websites (see Appendix C for details). Martino et al. [65] conducted a similar study and could convert IP addresses to their associated domains for the Tranco top 6K websites with an accuracy of 50.5%. Patil et al. [81] conduct a one-off measurement study to examine the uniqueness of IP-based fingerprints and find that 95.7% of websites have a unique fingerprint. However, similar to most prior WF studies, they do not consider the impact of caching while also lacking the temporal aspect of fingerprints. In practice, these essential factors cannot be neglected because the dynamics of web contents [15, 57] and domain–IP mappings over time [45] can impact the fingerprints [64]. We address these shortcomings by not only taking browser caching into consideration but also conducting our measurement in a longitudinal fashion to investigate the effectiveness of our fingerprints over time.

12 Conclusion

Domain name encryption is an important and necessary step to safeguard the domain name information, which is still largely being transferred in an unsecured manner on the Internet. However, we have shown that encryption alone is not enough to protect web users' privacy. Especially when it comes to preventing nosy network observers from tracking users' browsing activities, the IP address information of remote servers being contacted is still visible, which can then be employed to infer the visited websites.

In this study, we construct IP-based fingerprints for more than 200K websites by performing active DNS measurement to periodically resolve the contacted domain names while visiting these websites. Using these IP-based fingerprints, we could successfully identify 84% of the websites based solely on the IP addresses observed from the network traffic. Even when browser caching or ad blocking is considered, reducing the network traffic an on-path adversary can observe, our fingerprinting technique can still identify 80% of the websites studied.

Our findings show that significant effort still needs to be invested by both website owners and hosting providers to maximize the privacy benefits offered by domain name encryption. Specifically, website owners should try to minimize references to web resources loaded from domains other than their website's primary domain, and refrain from hosting their website on servers that do not co-host any other websites. Hosting providers can help to hinder IP-based WF by co-locating many websites on the same server(s), while also dynamically changing mappings between domains and their hosting IPs.

Acknowledgments

We would like to thank our shepherd, Tobias Pulls, and the anonymous reviewers for their thorough feedback on earlier drafts of this paper. We also thank Shachee Mishra, Tapti Palit, Seyedhamed Ghavamnia, Jarin Firose Moon, Md Mehedi Hasan, Kien Huynh, Thang Bui, Huan Nguyen, and Thai Le for helpful conversations and suggestions.

This research was supported in part by the Open Technology Fund under an Information Controls Fellowship. The opinions in this paper are those of the authors and do not necessarily reflect the opinions of the sponsor.

References

[1] Pre-alpha: Run an Onion Proxy Now! https://lists.torproject.org/pipermail/tor-dev/2002-September/002374.html.

[2] Cisco IOS NetFlow. http://bit.ly/CiscoNetFlow, 2012.

[3] Encrypt the Web. https://eff.org/encrypt-the-web, 2019.

[4] Cloudflare DoH. http://bit.ly/CloudflareDoH, 2020.

[5] Quantcast. https://www.quantcast.com/top-sites/, 2020.

[6] Stat Counter: Browser Market Share Worldwide. https://gs.statcounter.com/browser-market-share, 2020.

[7] State of the Web. https://httparchive.org/reports/state-of-the-web, 2020.

[8] Verisign report - the domain name industry brief. https://bit.ly/Verisign-Report, 2020.

[9] Alexa Top Sites. https://www.alexa.com/, 2021.

[10] IP Feeds by FireHOL. https://iplists.firehol.org/, 2021.

[11] The Majestic Million. http://bit.ly/MajesticList, 2021.

[12] Umbrella popularity list. http://bit.ly/UmbrellaList, 2021.

[13] Azadeh Akbari and Rashid Gabdulhakov. Platform Surveillance and Resistance in Iran and Russia : The Case of Telegram. In *Surveillance and Society*, 2019.

[14] Anonymous, AA. Niaki, NP. Hoang, P. Gill, and A. Houmansadr. Triplet censors: Demystifying great firewall's DNS censorship behavior. In *USENIX FOCI '20*.

[15] Ricardo Baeza-Yates, Carlos Castillo, and Felipe Saint-Jean. *Web Dynamics, Structure, and Page Quality*. 2004.

[16] Robert Beverly. A Robust Classifier for Passive TCP/IP Fingerprinting. In *PAM '04*.

[17] Simon Blake-Wilson, Magnus Nystrom, David Hopwood, Jan Mikkelsen, and Tim Wright. Transport Layer Security (TLS) Extensions. RFC 3546, IETF, June 2003.

[18] Thomas Brewster. Now Those Privacy Rules Are Gone, This Is How ISPs Will Actually Sell Your Personal Data. https://bit.ly/Forbes-ISP-sells-data, 2017.

[19] J. Bushart and C. Rossow. Padding ain't enough: Assessing the privacy guarantees of encrypted DNS. In *FOCI '20*.

[20] X. Cai, R. Nithyanand, T. Wang, R. Johnson, and I. Goldberg. A Systematic Approach to Developing and Evaluating Website Fingerprinting Defenses. In *ACM CCS '14*.

[21] Frank Cangialosi, Taejoong Chung, David Choffnes, Dave Levin, Bruce M. Maggs, Alan Mislove, and Christo Wilson. Measurement and analysis of private key sharing in the https ecosystem. In *ACM CCS '16*.

[22] Giovanni Cherubin, Jamie Hayes, and Marc Juarez. Website Fingerprinting Defenses at the Application Layer. 2017.

[23] S. Coull, M. Collins, C. Wright, F. Monrose, and M. Reiter. On web browsing privacy in anonymized netflows. In *USENIX Security '07*.

[24] W. Cui, T. Chen, C. Fields, J. Chen, A. Sierra, and E. Chan-Tin. Revisiting assumptions for website fingerprinting attacks. In *ACM AsiaCCS '19*.

[25] Casey Deccio and Jacob Davis. DNS Privacy in Practice and Preparation. In *ACM CoNEXT '19*.

[26] R. Dingledine, N. Mathewson, and P. Syverson. Tor: The Second-Generation Onion Router. In *USENIX Security '04*.

[27] H. Duan, N. Weaver, Z. Zhao, M. Hu, J. Liang, J. Jiang, K. Li, and V. Paxson. Hold-On: Protecting Against On-Path DNS Poisoning. In *SATIN '12*.

[28] AP. Felt, R. Barnes, A. King, C. Palmer, C. Bentzel, and P. Tabriz. Measuring HTTPS Adoption on the Web. In *USENIX Security '17*.

[29] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, and T. Berners-Lee. HTTP/1.1. RFC 2616, June 1999.

[30] Edward W. Forgy. Cluster Analysis of Multivariate Data : Efficiency Versus Interpretability of Classifications. 1965.

[31] Christian Fuchs, Kees Boersma, Anders Albrechtslund, and Marisol Sandoval. Internet and Surveillance: The Challenges of Web 2.0 and Social Media. 2011.

[32] I. Goldberg, D. Wagner, and E. Brewer. Privacy-Enhancing Technologies for the Internet. In *Proceedings of the 42nd IEEE International Computer Conference*, 1997.

[33] J. Gong and T. Wang. Zero-delay Lightweight Defenses against Website Fingerprinting. In *USENIX Security '20'*.

[34] R. Gonzalez, Claudio Soriente, and Nikolaos Laoutaris. User profiling in the time of https. In *ACM IMC '16*.

[35] Google. JSON API for DNS over HTTPS (DoH). https://developers.google.com/speed/public-dns/docs/dns-over-https, 2019.

[36] Google Developers. Remove Render-Blocking JavaScript. https://developers.google.com/speed/docs/insights/BlockingJS, 2018.

[37] B. Greschbach, T. Pulls, LM. Roberts, P. Winter, and N. Feamster. The Effect of DNS on Tor's Anonymity. In *NDSS '17*.

[38] Ilya Grigorik. Critical Rendering Path. http://bit.ly/CriticalRenderingPath, 2018.

[39] B. Haas. Man in China Sentenced to Five years' Jail for Running VPN. https://www.theguardian.com/world/2017/dec/22/man-in-china-sentenced-to-five-years-jail-for-running-vpn.

[40] J. Hayes and G. Danezis. k-fingerprinting: A Robust Scalable Website Fingerprinting Technique. In *USENIX Security Symposium 2016*.

[41] A. Hintz. Fingerprinting Websites Using Traffic Analysis. In *Conference on Privacy Enhancing Technologies*, 2002.

[42] NP. Hoang, Y. Asano, and M. Yoshikawa. Your Neighbors Are My Spies: Location and other Privacy Concerns in GLBT-focused Location-based Dating Applications. In *Trans. on Advanced Communications Technology 2016*.

[43] NP. Hoang, P. Kintis, M. Antonakakis, and M. Polychronakis. An Empirical Study of the I2P Anonymity Network and Its Censorship Resistance. In *ACM IMC '18*.

[44] NP. Hoang, I. Lin, S. Ghavamnia, and M. Polychronakis. K-resolver: Towards Decentralizing Encrypted DNS Resolution. In *MADWeb '20*.

[45] NP. Hoang, AA. Niaki, N. Borisov, P. Gill, and M. Polychronakis. Assessing the Privacy Benefits of Domain Name Encryption. In *ACM AsiaCCS '20*.

[46] NP. Hoang, AA. Niaki, J. Dalek, J. Knockel, P. Lin, B. Marczak, M. Crete-Nishihata, P. Gill, and M. Polychronakis. How Great is the Great Firewall? Measuring China's DNS Censorship. In *USENIX Security '21*.

[47] NP. Hoang, AA. Niaki, M. Polychronakis, and P. Gill. The Web is Still Small After More Than a Decade. *ACM SIGCOMM Computer Communication Review 2020*.

[48] NP. Hoang and D. Pishva. Anonymous Communication and Its Importance in Social Networking. In *ICACT '14*.

[49] P. Hoffman and P. McManus. DNS queries over HTTPS. RFC 8484, IETF, October 2018.

[50] Rebekah Houser, Zhou Li, Chase Cotton, and Haining Wang. An Investigation on Information Leakage of DNS over TLS. In *ACM CoNEXT*, 2019.

[51] Z. Hu, L. Zhu, J. Heidemann, A. Mankin, D. Wessels, and P. Hoffman. Specification for DNS over transport layer security (TLS). RFC 7858, IETF, May 2016.

[52] Kevin Jacobs. Encrypted Client Hello: the future of ESNI in Firefox. http://blog.mozilla.org/security/2021/01/07/encrypted-client-hello-the-future-of-esni-in-firefox, 2021.

[53] Marc Juarez, Sadia Afroz, Gunes Acar, Claudia Diaz, and Rachel Greenstadt. A Critical Evaluation of Website Fingerprinting Attacks. In *ACM CCS*, 2014.

[54] Marc Juarez, Mohsen Imani, Mike Perry, Claudia Diaz, and Matthew Wright. Toward an efficient website fingerprinting defense. In *ESORICS*, 2016.

[55] Sarah Krouse and Patience Haggin. Internet Providers Look to Cash In on Your Web Habits. https://www.wsj.com/articles/facebook-knows-a-lot-about-you-so-does-your-internet-provider-11561627803, 2019.

[56] Douglas J. Leith. Web browser privacy: What do browsers say when they phone home? 2020.

[57] Mark Levene. *Web dynamics: Adapting to change in content, size, topology and use*. Springer Science & Business Media, 2004.

[58] M. Liberatore and BN. Levine. Inferring the Source of Encrypted HTTP Connections. In *ACM CCS '06*, 2006.

[59] T. Libert and R. Binns. Good news for people who love bad news: Centralization, privacy, and transparency on us news sites. *ACM Conference on Web Science*, 2019.

[60] Stuart P. Lloyd. Least squares quantization in pcm. 1982.

[61] Chaoyi Lu, Baojun Liu, Zhou Li, Shuang Hao, Haixin Duan, Mingming Zhang, Chunying Leng, Ying Liu, Zaifeng Zhang, and Jianping Wu. An End-to-End, Large-Scale Measurement of DNS-over-Encryption: How Far Have We Come? In *ACM Internet Measurement Conference*, 2019.

[62] X. Luo, P. Zhou, E. Chan, W. Lee, R. Chang, and R. Perdisci. HTTPOS: Sealing Information Leaks with Browser-side Obfuscation of Encrypted Flows. In *Network and Distributed System Security Symposium*, 2011.

[63] James B. MacQueen. Some methods for classification and analysis of multivariate observations. 1967.

[64] M. Di Martino, P. Quax, and W. Lamotte. Realistically Fingerprinting Social Media Webpages in HTTPS Traffic. In *ACM ARES '19*.

[65] Mariano Di Martino, P. Quax, and W. Lamotte. Knocking on IPs: Identifying HTTPS Websites for Zero-Rated Traffic. *Security and Communication Networks 2020*.

[66] A. Mayrhofer. Padding Policies for EDNS(0). RFC 8467, IETF, 2018.

[67] Patrick McManus. Improving DNS privacy in firefox. https://blog.nightly.mozilla.org/2018/06/01/improving-dns-privacy-in-firefox/, 2018.

[68] MDN Web Docs. Domain sharding. https://developer.mozilla.org/en-US/docs/Glossary/Domain_sharding, 2020.

[69] MDN Web Docs. DOMContentLoaded event. https://developer.mozilla.org/en-US/docs/Web/API/Window/DOMContentLoaded_event, 2020.

[70] Brad Miller, Ling Huang, Anthony D. Joseph, and J. Doug Tygar. I Know Why You Went to the Clinic: Risks and Realization of HTTPS Traffic Analysis. In *Privacy Enhancing Technologies Symposium*, 2014.

[71] Alec Muffett. No Port 53, Who Dis? A year of DNS over HTTPS over Tor. In *DNS Privacy Workshop 2021*.

[72] Rayan Naqash. India's crackdown on VPNs in Kashmir seeks to quell cyber-insurgency threat but risks blowback. hhttps://bit.ly/India-blocks-VPN, 2020.

[73] Milad Nasr, Alireza Bahramali, and Amir Houmansadr. DeepCorr: Strong Flow Correlation Attacks on Tor Using Deep Learning. In *ACM CCS*, 2018.

[74] Milad Nasr, Amir Houmansadr, and A. Mazumdar. Compressive traffic analysis: A new paradigm for scalable traffic analysis. In *ACM CCS '17*.

[75] Arian Akhavan Niaki, Shinyoung Cho, Zachary Weinberg, Nguyen Phong Hoang, Abbas Razaghpanah, Nicolas Christin, and Phillipa Gill. ICLab: A Global, Longitudinal Internet Censorship Measurement Platform. In *Symposium on Security and Privacy*, May 2020.

[76] Nick Nikiforakis, Luca Invernizzi, Alexandros Kapravelos, Steven Van Acker, Wouter Joosen, Christopher Kruegel, Frank Piessens, and Giovanni Vigna. You are what you include: Large-scale evaluation of remote javascript inclusions. In *ACM Conference on Computer and Communications Security*, 2012.

[77] Rishab Nithyanand, Xiang Cai, and Rob Johnson. Glove: A bespoke website fingerprinting defense. In *WPES*, 2014.

[78] NP. Hoang and S. Doreen and M. Polychronakis. Measuring I2P Censorship at a Global Scale. In *FOCI '19*.

[79] Andriy Panchenko, Fabian Lanze, Jan Pennekamp, Thomas Engel, Andreas Zinnen, Martin Henze, and Klaus Wehrle. Website fingerprinting at internet scale. In *Network and Distributed System Security Symposium*, 2016.

[80] Andriy Panchenko, Lukas Niessen, Andreas Zinnen, and Thomas Engel. Website fingerprinting in onion routing based anonymization networks. In *WPES*, 2011.

[81] S. Patil and N. Borisov. What Can You Learn from an IP? In *Applied Networking Research Workshop*, 2019.

[82] Paul Pearce, Ben Jones, Frank Li, Roya Ensafi, Nick Feamster, Nick Weaver, and Vern Paxson. Global Measurement of DNS Manipulation. In *USENIX Security '17*, 2017.

[83] Mike Perry. A Critique of Website Traffic Fingerprinting Attacks, 2013. https://blog.torproject.org/critique-website-traffic-fingerprinting-attacks.

[84] Victor Le Pochat, Tom Van Goethem, and Wouter Joosen. Evaluating the long-term effects of parameters on the characteristics of the tranco top sites ranking. In *USENIX Workshop on Cyber Security Experimentation and Test*, 2019.

[85] Tobias Pulls and Rasmus Dahlberg. Website fingerprinting with website oracles. *PETS*, 2020.

[86] Abbas Razaghpanah, Rishab Nithyanand, Narseo Vallina-Rodriguez, Srikanth Sundaresan, Mark Allman, Christian Kreibich, and Phillipa Gill. Apps, trackers, privacy, and regulators: A global study of the mobile tracking ecosystem. In *Network and Distributed System Security Symposium*, 2018.

[87] E. Rescorla, K. Oku, N. Sullivan, and C. Wood. ESNI for TLS 1.3. Internet draft, IETF, March 2020.

[88] E. Rescorla, K. Oku, N. Sullivan, and C. Wood. TLS Encrypted Client Hello. Internet draft, IETF, June 2020.

[89] Walter Rweyemamu, Christo Lauinger, Tobiasand Wilson, William Robertson, and Engin Kirda. Clustering and the Weekend Effect: Recommendations for the Use of Top Domain Lists in Security Research. In *PAM*, 2019.

[90] S. Santesson, M. Myers, R. Ankney, A. Malpani, S. Galperin, and C. Adams. X.509 Internet Public Key Infrastructure Online Certificate Status Protocol - OCSP. RFC 6960, IETF, June 2013.

[91] Mahrud Sayrafi. Introducing DNS resolver for Tor. https://blog.cloudflare.com/welcome-hidden-resolver/, 2018.

[92] Paul Schmitt, Anne Edmundson, Allison Mankin, and Nick Feamster. Oblivious DNS: Practical Privacy for DNS Queries. In *PETS*, 2019.

[93] Zain Shamsi, Ankur Nandwani, D. Leonard, and D. Loguinov. Hershel: Single-packet os fingerprinting. *IEEE/ACM Transactions on Networking*, 24:2196–2209, 2016.

[94] C. E. Shannon. A mathematical theory of communication. *SIGMOBILE Mob. Comput. Commun. Rev.*, 2001.

[95] Craig A. Shue, Andrew J. Kalafut, and Minaxi Gupta. The Web is Smaller Than It Seems. In *IMC'07*.

[96] Sandra Siby, Marc Juarez, Claudia Diaz, Narseo Vallina-Rodriguez, and Carmela Troncoso. Encrypted DNS => Privacy? A Traffic Analysis Perspective. In *NDSS '20*.

[97] S. Singanamalla, Suphanat Chunhapanya, Marek Vavrusa, Tanya Verma, P. Wu, Marwan Fayed, K. Heimerl, N. Sullivan, and C. Wood. Oblivious dns over https (odoh): A practical privacy enhancement to dns. In *DNS Privacy Workshop 2021*.

[98] Payap Sirinam, Mohsen Imani, Marc Juarez, and Matthew Wright. Deep fingerprinting: Undermining website fingerprinting defenses with deep learning. In *ACM Conference on Computer and Communications Security*, 2018.

[99] Qixiang Sun, Daniel R. Simon, Yi-Min Wang, Wilf Russell, Venkata N. Padmanabhan, and Lili Qiu. Statistical identification of encrypted web browsing traffic. In *IEEE Symposium on Security and Privacy*, 2002.

[100] Janos Szurdi, Balazs Kocso, Gabor Cseh, Jonathan Spring, Mark Felegyhazi, and Chris Kanich. The Long "Taile" of Typosquatting Domain Names. In *USENIX Security '14*.

[101] B. Trammell, A. Wagner, and B. Claise. Flow Aggregation for the IP Flow Information Export (IPFIX) Protocol. RFC 7015, IETF, Sep 2013.

[102] Martino Trevisan, Idilio Drago, Marco Mellia, and Maurizio M. Munafò. Towards web service classification using addresses and dns. In *IWCMC*, 2016.

[103] Martino Trevisan, Francesca Soro, M. Mellia, I. Drago, and Ricardo Morla. Does domain name encryption increase users' privacy? *ACM SIGCOMM Computer Communication Review*, 50:16 – 22, 2020.

[104] V. Le Pochat and T. Van Goethem and S. Tajalizadehkhoob and M. Korczyński and W. Joosen. Tranco: A Research-Oriented Top Sites Ranking Hardened Against Manipulation. In *NDSS '19*.

[105] Nino Vincenzo Verde, G. Ateniese, E. Gabrielli, L. Mancini, and A. Spognardi. No nat'd user left behind: Fingerprinting users behind nat from netflow records alone. *2014 IEEE 34th International Conference on Distributed Computing Systems*, pages 218–227, 2014.

[106] David Wagner and Bruce Schneier. Analysis of the ssl 3.0 protocol. In *Workshop on Electronic Commerce*, 1996.
[107] Tao Wang. High precision open-world website fingerprinting. In *IEEE S&P*, 2020.
[108] Tao Wang, Xiang Cai, Rishab Nithyanand, Rob Johnson, and Ian Goldberg. Effective attacks and provable defenses for website fingerprinting. In *USENIX Security*, 2014.
[109] Tao Wang and Ian Goldberg. Walkie-talkie: An efficient defense against passive website fingerprinting attacks. In *USENIX Security*, 2017.
[110] Xiao Sophia Wang, Aruna Balasubramanian, Arvind Krishnamurthy, and David Wetherall. How speedy is SPDY? In *USENIX NSDI*, 2014.
[111] Yixiao Xu, Tao Wang, Qi Li, Qingyuan Gong, Yang Chen, and Yong Jiang. A Multi-Tab Website Fingerprinting Attack. In *ACSAC*, 2018.
[112] Sophia Yang. China to ban online gaming, chatting with foreigners outside Great Firewall: Report. https://www.taiwannews.com.tw/en/news/3916690, 2020.
[113] zzz and Lars Schimmer. Peer Profiling and Selection in the I2P Anonymous Network. In *PET-CON*, 2009.

A Dynamics of Domain-IP Mapping

In this appendix, we analyze the mappings between domains and IP addresses observed throughout the whole study period to examine the dynamics of domain–IP mappings in today's web ecosystem. Over the two-month period, we observed 531K domain names, resulting in 693K unique IP address. Of these domains, 212K belong to the primary domain group selected in §5, and 319K are secondary domains. In total, we have gathered more than 7M domain–IP mappings.

Figure 8 shows the longevity analysis of domain–IP mappings of the two domain groups. More than 60% of the mappings in both groups last for less than a week (i.e., observed in no more than three consecutive data batches). In contrast, only 15% of primary and less than 5% of secondary domain mappings can be observed for the whole two-month period of our study. The high churn rate of most mappings after a week is one of the reasons why our IP-based fingerprints deteriorate after ten days since being constructed (§7).

However, the picture changes completely when examining the number of domains and IP addresses in each mapping group. We refer to the group of mappings that are observed in no more than three consecutive data batches as *dynamic mappings*, and to mappings that are observed continuously for the whole period of study as *static mappings*. Table 3 shows the breakdown of the number of unique domains and IP addresses observed in each mapping group. We can see two unbal-

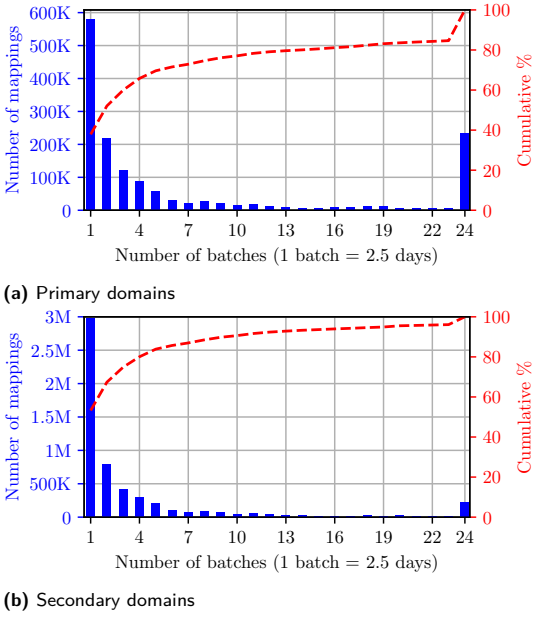

Fig. 8. Longevity of mappings between domains and their IPs.

anced allocations between (a) the total number of primary and secondary domain mappings, and (b) the dynamic and static mappings within the primary domains.

For (a), the number of secondary domain mappings is almost four times larger than primary domains, due to the fact that a visit to a primary domain loads several secondary domains. For (b), there are only 36K (17%) primary domains with a high IP address churn rate, occupying a pool of 169K unique IP addresses. In contrast, 167K (79%) primary domains remain stable on the same IP addresses for the whole period of our study. This explains the reason why many of our IP-based fingerprints are still effective after two months (§7). Specifically, although 50% of the IP addresses are changed in more than 50% of the fingerprints, as shown in Figure 4, this is mainly due to the change of secondary domains' hosting IP addresses. On the other hand, after two months, almost 80% of primary domains are still hosted on static IP addresses, contributing to the validity of our IP-based fingerprints.

Although only a small number of domains whose hosting IP addresses are changed frequently, our findings show that it is totally possible to increase the dynamics of domain–IP mappings from the perspective of hosting providers. One may consider that frequently changing the hosting IP addresses is not feasible for those web servers that use "cruise-liner" certificates [21], in which numerous domains are aggregated in each certificate to support non-SNI clients. However, to the best of our knowledge, the use of "cruise-liner" cer-

Table 3. Breakdown of the number of domains and IPs in dynamic and static mappings.

Domain & IP type	Total mappings	Dynamic mappings	Static mappings
Primary domains	1.5M	36K (17%)	167K (79%)
Primary IPs		169K (45%)	137K (36%)
Secondary domains	5.5M	208K (65%)	88K (28%)
Secondary IPs		306K (69%)	77K (17%)

tificates has been deprecated by most hosting providers due to widespread support of SNI by all major browsers. For instance, while Cloudflare used to employ "cruise-liner" certificates for websites co-hosted on its CDNs, the Subject Alternative Names field of Cloudflare's certificates now contains only the websites' domain and `sni.cloudflaressl.com`. Thus, our suggestion is still compatible with multiple certificates per IP address. In fact, Cloudflare does allow website owners to upload their own certificates instead of using Cloudflare's.

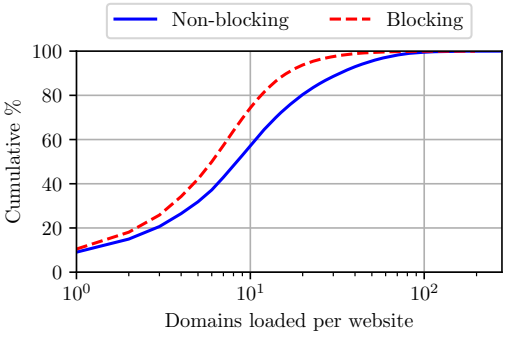

Fig. 9. CDF of domain names loaded per website.

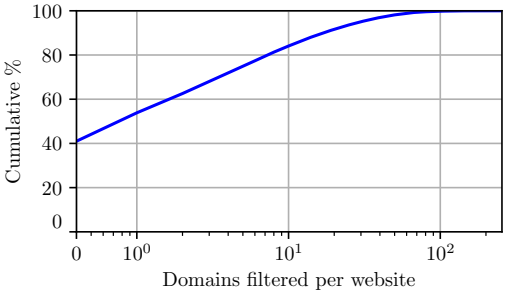

Fig. 10. CDF of domains filtered per website by Brave.

B Domains Filtered by Brave

While the accuracy rate does decrease when fingerprinting against websites browsed with Brave (§8.2), our basic fingerprinting approach could still obtain a relatively high accuracy rate of 80%. To that end, we conduct an additional analysis on the filtered domains to find the underlying reason why removing these resources does not substantially reduce websites' fingerprintability.

As shown in Figure 9, the number of domains loaded per website when browsing with Brave (dashed line) is significantly lower than when using a non-blocking browser (solid line). Specifically, almost 80% of websites load less than ten domains with Brave, whereas only 57% of websites load less than ten domains using a non-blocking browser. The average number of domains loaded per website with Brave is only eight, whilst there are 14 domains loaded per website on average for a non-blocking browser. Of all websites studied, 41% of websites do not have any domains filtered by the Brave browser. The remaining 59% of these websites have at least one domain filtered, as shown in Figure 10.

Table 4 shows the top-ten most blocked domains by Brave, with `www.google-analytics.com` being the most blocked domain. Among the 220K websites studied, it is removed from 69K (31%) websites. Although the domain is referenced in more than half of the websites (as shown in Table 1), Brave does not entirely remove it from all of them, depending on how it is referenced on each website. Despite being removed from a large number of websites, as indicated in the third column of the table, these domains only contribute a small amount of information entropy to the fingerprint when included (as discussed in §6.1). This is the reason why our fingerprinting technique can still identify a relatively high number (80%) of websites when browsing with Brave. Analyzing the MIME type of objects loaded from these domains, we find that the vast majority of them are used to load images and scripts used for tracking and advertisement services operated by Google and Facebook.

C Impact of Co-location and Popularity on Attack Accuracy

We next analyze the co-location degree and popularity ranking of the fingerprinted websites to investigate whether there are any correlations between these properties of a given website and the chance that it can be precisely fingerprinted. Figure 11 shows two scatter plots of the popular websites and sensitive websites that we could successfully fingerprint, with respect to their co-location degree and popularity ranking. As expected, websites that are not co-hosted with any other

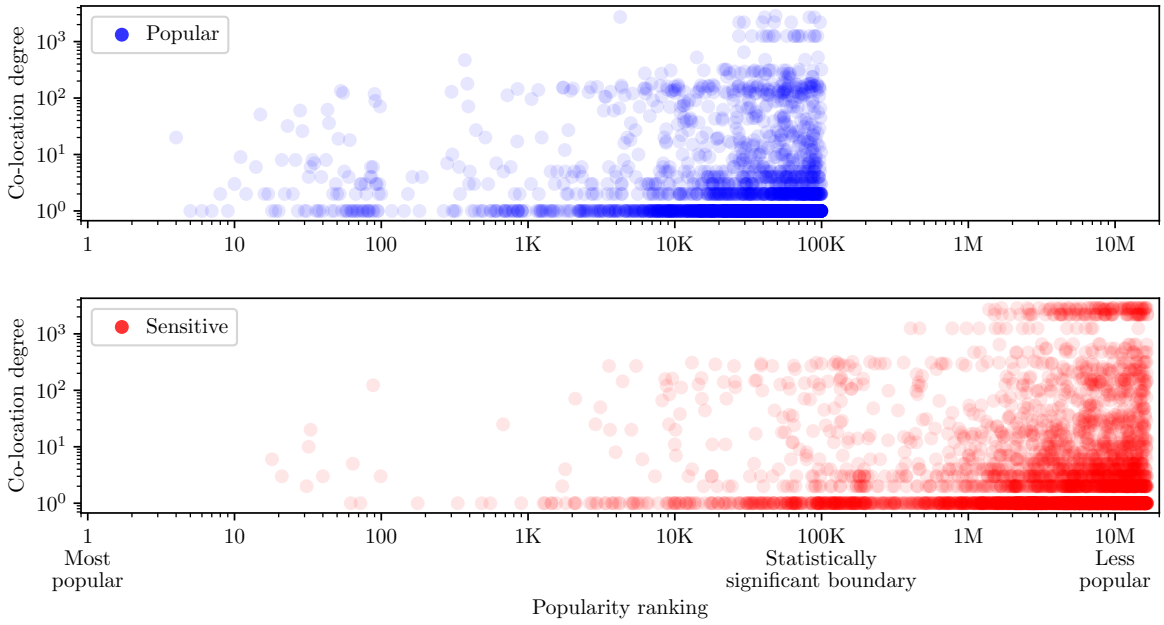

Fig. 11. Co-location degree vs. popularity ranking distribution of *successfully fingerprinted websites*. "Popular" corresponds to 90,231 fingerprinted websites from the Tranco list's top-100K most popular domains, while "Sensitive" corresponds to 104,983 fingerprinted websites from 126,597 sensitive domains chosen from Alexa's sensitive categories that span the whole ranking spectrum. 5,687 fingerprinted websites are common between the two sets.

Table 4. Top-ten domains removed by Brave.

Domain name	# Blocked	Entropy
www.google-analytics.com	69K (31%)	0.87
stats.g.doubleclick.net	57K (26%)	1.53
www.google.com	38K (17%)	1.44
www.googletagmanager.com	32K (15%)	1.71
www.facebook.com	31K (14%)	1.97
googleads.g.doubleclick.net	28K (13%)	2.10
tpc.googlesyndication.com	21K (10%)	3.08
connect.facebook.net	21K (10%)	1.98
adservice.google.com	18K (8%)	2.68
pagead2.googlesyndication.com	18K (8%)	3.03

websites are the most susceptible to our IP-based fingerprinting attack. Regardless of having a high co-location degree, however, websites can still be fingerprinted with our enhanced technique due to the inclusion of unique secondary domains.

Hoang et al. [45] suggest an ideal co-location threshold of at least 100 domains per hosting IP address, so that the co-hosted websites can gain some meaningful privacy benefit from the deployment of domain name encryption. However, Figure 11 shows that even when more than 100 websites are co-hosted, they can still be fingerprinted. Again, the underlying reason is that these websites often reference several external resources, making their fingerprint more distinguishable compared to the rest of the co-hosted websites.

In addition, Figure 12 shows the CDF of the popularity ranking of the successfully fingerprinted websites. We can see that the number of fingerprinted websites slightly leans towards more popular rankings, which can also be confirmed by the higher accuracy rates when it comes to fingerprinting the popular websites compared to the sensitive websites, as indicated in all three fingerprinting approaches in Table 2.

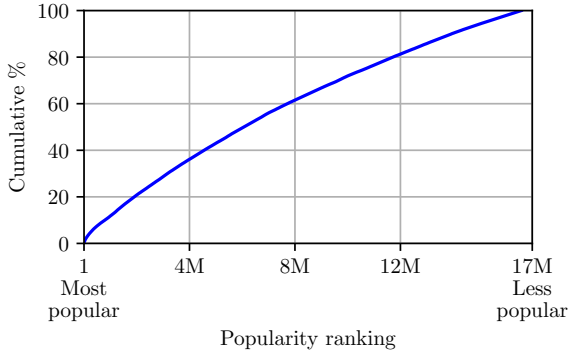

Fig. 12. CDF of popularity ranking as a percentage of all successfully fingerprinted websites.