What Can You Learn from an IP?

Simran Patil
University of Illinois at Urbana-Champaign
sppatil2@illinois.edu

Nikita Borisov
University of Illinois at Urbana-Champaign
nikita@illinois.edu

ABSTRACT

The Internet was not designed with security in mind. A number of recent protocols such as Encrypted DNS, HTTPS, etc. target encrypting critical parts of the web architecture, which can otherwise be exploited by eavesdroppers to infer users' data. But encryption may not necessarily guarantee privacy, especially when it comes to metadata. Emerging standards can protect the contents of both DNS queries and the TLS SNI extensions; however, it might still be possible to determine which websites users are visiting by simply looking at the destination IP addresses on the traffic originating from users' devices. We perform a measurement study to determine the anonymity provided by IP addresses resulting from the multiple sub-queries that are made as a consequence of accessing a particular web page. We show that, in most cases, an adversary can use the IP addresses during a page load as a form of a fingerprint to infer the original site identity.

ACM Reference Format:
Simran Patil and Nikita Borisov. 2019. What Can You Learn from an IP?. In *ANRW '19: Applied Networking Research Workshop (ANRW '19), July 22, 2019, Montreal, QC, Canada.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3340301.3341133

1 INTRODUCTION

The growing deployment of HTTPS [2] means that the web browsing traffic of most users is commonly well protected. Yet while plaintext HTTP is on the decline, DNS, another protocol critical to web browsing, continues to be unencrypted. DNS has the potential to leak large amounts of sensitive information, in particular the identities of site you are visiting, to an array of network observers. Recently, however, IETF has developed encrypted versions of the DNS protocol. Most widely deployed are DNS-over-TLS and DNS-over-HTTPS [1, 6], which protect the contents of both DNS queries and responses from eavesdroppers by relying on existing deployed end-to-end encryption protocols.[1] In the context of web browsing, however, a DNS request is followed by an HTTP or HTTPS connection to a web server located by the request. In the case of HTTP, the entire request is sent in plaintext. With HTTPS, TLS protects the majority of the communication section, yet some data is sent in the clear. Most importantly, the server name indication (SNI) extension [7] specifies the domain name of the web server, which is used to select the appropriate certificate for the TLS handshake. Essentially, even with encrypted DNS, the queried domain is then immediately sent in the clear in the SNI and the certificate!

To address this problem, starting with TLS version 1.3 [11], the certificate messages are encrypted. Additionally, a draft encrypted SNI extension [12] provides a mechanism to remove the last plaintext indicator of the domain name from the network, leaving an observer with just an IP address. In this paper, we start to study the question of how much information the IP address reveals about the soon-to-be-hidden domain name.

We adopt the model of an adversary who aims to recover domain information by collecting forward mappings of various candidate domains, and then using the answers to infer the reverse mapping of a given IP. This can be done by collecting popular domains from a data set such as the Alexa top lists [14], Chrome User Experience Report [5] or certificate transparency lists [9]. While none of these provides a comprehensive list of potential domains a user might visit, these lists cover a large fraction of DNS lookups during typical web browsing.

An observer may further make use of the fact that a typical web page will cause dozens of objects to be loaded from a number of different web servers. The set of all IPs contacted by a page load constitutes a *page load fingerprint* (PLF), which

ANRW '19, July 22, 2019, Montreal, QC, Canada

ACM ISBN 978-1-4503-6848-3/19/07…$15.00
https://doi.org/10.1145/3340301.3341133

[1] Note that in the context of encrypted DNS, not the entirety of the DNS resolution process is encrypted. Instead, the user connects to a recursive resolver using end-to-end encryption, while the resolver will typically use plaintext DNS queries to contact authoritative servers. The queries *are* hidden from a local network observer, and someone observing traffic from the recursive resolver may not necessarily be able to associate it with a given users. Encrypted DNS does not prevent the resolver from learning which domains the user is visiting; it may be possible to mitigate this by connecting to the resolver over Tor or similar anonymity service [13].

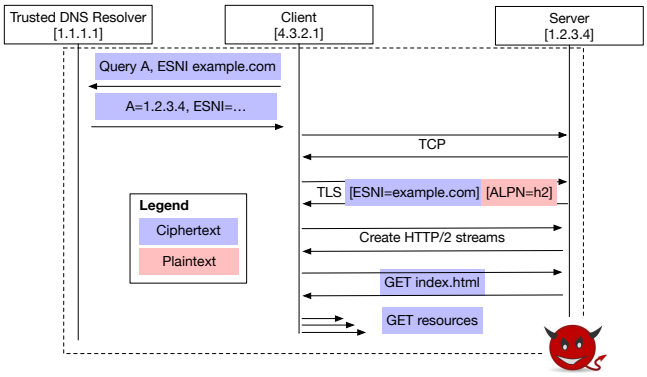

Figure 1: Sample view of website fingerprinting adversary. The adversary is assumed to see the entirety of the traffic originated by the client, but TLS1.3, encrypted DNS, and ESNI ensure that all but the metadata except for the IP address are encrypted.

can be used to distinguish different sites even if their main origin domain uses an IP shared with different sites.

We therefore perform a measurement study to determine the anonymity provided by each single IP address, as well as the anonymity of page load fingerprints. We use data from an instrumented web browser that visited the top 1 million sites as ranked by Alexa [14] and analyze the anonymity sets. Our initial findings show that while some pages provide a certain degree of anonymity, in the vast majority of the cases, a web page has a unique page load fingerprint. This means that effective protection of domain metadata associated with web browsing will require not only protocol changes that remove the overt domain information but also changes to web hosting infrastructure to prevent inferences.

2 PRELIMINARIES

2.1 Background and Threat Model

Website page loads often begin with a user action, such as typing in a website address in a browser address bar or clicking on a link. Such events typically induce several distinct network events, such as resolving domain names to IP addresses with DNS and then subsequently creating HTTP(S) connections to fetch page resources. Often, rendering a single page requires loading several subresources, such as scripts, style sheets, and images. For example, to load the front page of `nytimes.com`, browsers will first send a `GET` request for `index.html`, which contains references to first or third party resources (style sheets, Javascript files, images, etc.) necessary to render the entire page. The browser will then recursively fetch these resources, possibly by performing additional DNS queries and creating new HTTPS connections.

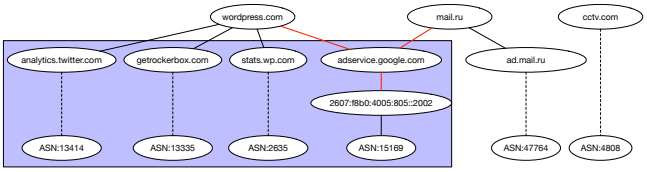

Figure 2: Minimized domain tree for `wordpress.com`, `mail.ru`, and `cctv.com`.

The addresses associated with connections made or used while fetching subresources constitute a "page load fingerprint" (PLF). Each subresource connection is associated with one or more IP addresses, each of which in turn is associated with an Autonomous System (AS). As an example, consider the "domain tree" shown in Figure 2. The page `wordpress.com` has subresources that are ultimately serviced by a variety of ASs (those in the blue box). It has one subresource connection that is also shared by the `mail.ru` page. The `cctv.com` page has no shared subresource connections. Considering only these connections, a single connection to `adservice.google.com` could refer to a page load of `wordpress.com` or `mail.ru`. However, if one observes the entire set of connections in the blue box, i.e., the PLF, then they most likely are for `wordpress.com` rather than `mail.ru`.

The goal of an address-based website fingerprinting attack is to identify the target of a page load by examining PLFs. Note that this is not the same threat model of encrypted DNS or SNI, wherein the goal is to learn the destination of a *specific* connection.

Our threat model encompasses a *local* adversary $\mathcal{A}$ capable of observing all outgoing IP packets between victim clients and servers, as shown in Figure 1. Thus, $\mathcal{A}$ can monitor the set of TLS connections from a client to any server, and use this information to fingerprint connections. We assume secure DNS and ESNI both (reliably) protect this information from passive observers.[2] Session-layer information such as encrypted packet sizes, directions, and timing are out of scope.

2.2 Related Work

Network connections by nature reveal information about endpoint behavior. For example, a connection to 8.8.8.8 indicates usage of Google's DNS service. Likewise, a connection to any address in a Cloudflare IP address block indicates use of a service hosted by Cloudflare. The relationship between network address and domains, especially when stable and

[2]Note that various side channels such as request and response sizes, as well as cache timing side channels, may be used by $\mathcal{A}$ to reveal the contents of a DNS query. For this paper, we assume such attacks are infeasible and focus strictly on the network-layer connection information, such as IP addresses.

unique, are a strong signal for website fingerprinting. Trevisan et al. [16] explored use of this signal as a reliable mechanism for website fingerprinting. They find that most major services (domains) have clearly associated IP address(es), though these addresses may change over time. Jiang et al. [8] and Tammaro et al. [15] also previously came to the same conclusion. Thus, classifiers that rely solely on network addresses may be used to aid website fingerprinting attacks.

In this paper we restrict our feature set to that which we can obtain from HTTPS connections. Flow classifiers using other protocol messages and features such as DNS messages [4] and the TLS Server Name Indication (SNI) exist, though we assume technologies such as DNS-over-TLS [1], DNS-over-HTTPS [6], and Encrypted SNI [12] keeps this information "safe" by encryption.

3 MEASUREMENT STUDY

To evaluate the potential privacy benefits of encrypted DNS and SNI for a user who is browsing the web, we perform a measurement study by first crawling the most popular websites, as determined by Alexa [14], and then performing DNS resolution on all domains involved in rendering each website.

3.1 Data Set

We used a web measurement tool called MIDA, a highly configurable web crawler built on top of Chromium and the Chrome DevTools Protocol [10], to direct a Chromium browser to visit Alexa's top 1 Million websites [14] and fetch the information associated with these requests, which includes an in-depth summary of the browser metadata, resource information, details about sub-query URLs, resources hosted, etc. table 1 shows the result of our MIDA crawl on 1 million sites. Note that about 5% of sites experience a failure during our attempt to visit them, so our data set is slightly smaller than a million. On average, each site loaded approximately 96 different URLs from 16.5 different domains. The web crawl was performed in late March 2019.

We then performed name resolution on all of the domains from the 90 million URLs involved in the web crawl using zdns[3], a high-performance bulk DNS resolution tool. We used zdns in iterative mode, bypassing our local resolver and its cache. DNS results are well known to vary across both time and location; all of our lookups were performed from a single vantage point at the University of Illinois within a roughly two-hour time span. We note that an adversary could use tools such as zdns to collect similar data sets, using a vantage point that matches that of the victim. In addition to looking up the IP addresses corresponding to each domain, we tracked the sequence of CNAME redirections, which we used to attribute domains to content distribution networks (CDNs) using heuristics from the cdnfinder tool [17]. Note that this attribution is imperfect, since many CDNs do not utilize descriptive CDNs. We also perform reverse DNS lookup (PTR) for each of the IP addresses returned by forward lookups.

[3]https://github.com/zmap/zdns

Table 1: Web Crawl Data based on Alexa top 1M sites. 944 094 sites were successfully loaded.

Request type	Count
Document (HTML)	4 162 754
Image	43 444 344
Script	23 379 941
Stylesheet	8 633 523
XHR	4 675 860
Font	4 051 221
Other	1 647 949
Fetch	347 730
Media	167 775
EventSource	1 718
Manifest	749
TextTrack	616
Total	90 514 000

Table 2: Resource types for requests that use unique domains (domains that resolve to an IP with anonymity set 1).

Request type	Count	Frac of all requests
Document (HTML)	554 753	13.3%
Image	7 347 703	16.9%
Script	3 234 593	13.8%
Stylesheet	1 262 243	14.6%
XHR	1 052 933	27.0%
Font	235 446	5.8%
Other	387 916	23.5%
Fetch	34 294	9.9%
Media	24 646	14.7%
EventSource	190	11.1%
Manifest	134	17.9%
TextTrack	21	3.4%
Total	14 134 872	15.6%

3.2 Single IP lookups

Among the 90 million object requests in our data set, there are 1 819 087 unique domain names. We were able to successfully resolve 1 795 506 (98.7%) of these, obtaining 741 049 distinct IP addresses. (Note that each domain name resolves to an

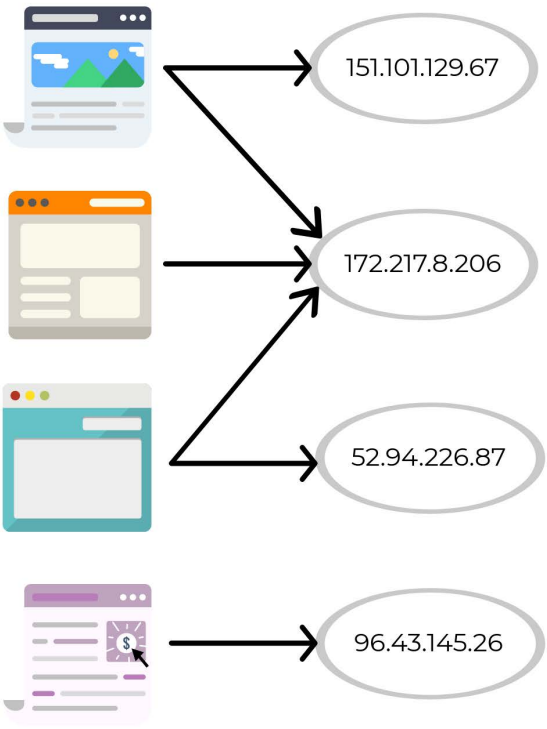

Figure 3: If an IP address is hit by several different web sites, then a reverse DNS lookup will not provide much information about which website the user was looking at. But if the IP address has a one-to-one backward mapping to a website then the chances of the user's web activity being profiled increase significantly which is a threat to the user's privacy.

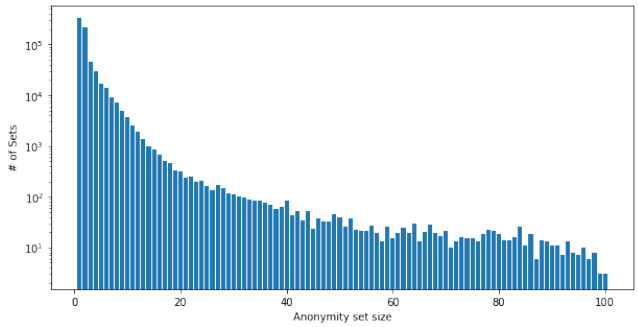

Figure 4: A histogram of IP anonymity set sizes. For each IP in our dataset we calculate the number of domains that map as its anonymity set. The median anonymity set has size 2, and the average is 3.14. The largest is 16 050 (only the top 100 are shown in the figure for clarity).

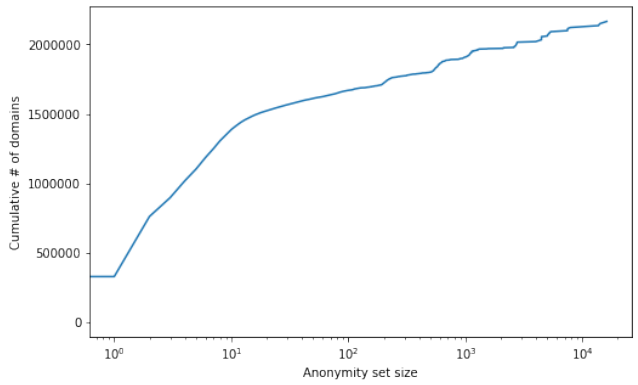

Figure 5: A CDF of the anonymity set size that domains map to.

average of 1.46 IP addresses, but many domain names map to the same address.)

We can now calculate how well an adversary, armed with this data set, can map an IP back to a domain name. For each IP address, we compute the set of domain names that map to it as its *anonymity set*. Figure 4 shows a histogram of these sizes. A slight minority of the IPs in our data set (47.6%) correspond to a single domain. For these domains, under our threat model, where the adversary knows the set of potential addresses a user may look up and is able to perform forward lookups on them, encrypted DNS provides little to no benefit. Note that this technique is much more successful than using reverse DNS—only 34 840 domains resolved to IPs that had the corresponding domain as its rDNS entry. The median anonymity set size is 2 and the average is 3.14. Some IP addresses map to a large set of addresses, including one that corresponds to over 16 000 domains.

Figure 5 shows a cumulative distribution function of the anonymity set sizes that each domain belongs to. Note that since larger anonymity set sizes have more domains, a median domain corresponds to an IP address of an anonymity set size of 4.

We do note that there is some potential for consolidation that is present here. We use the domain names returned during a lookup (including CNAMEs) to classify a sample of our IP addresses as belonging to various content distribution networks, as shown in fig. 6. This shows that a significant fraction of addresses come from CDNs. Today, many CDNs are able to serve a large number of sites from a small set of IP addresses (a feature exploited by domain fronting [3]).

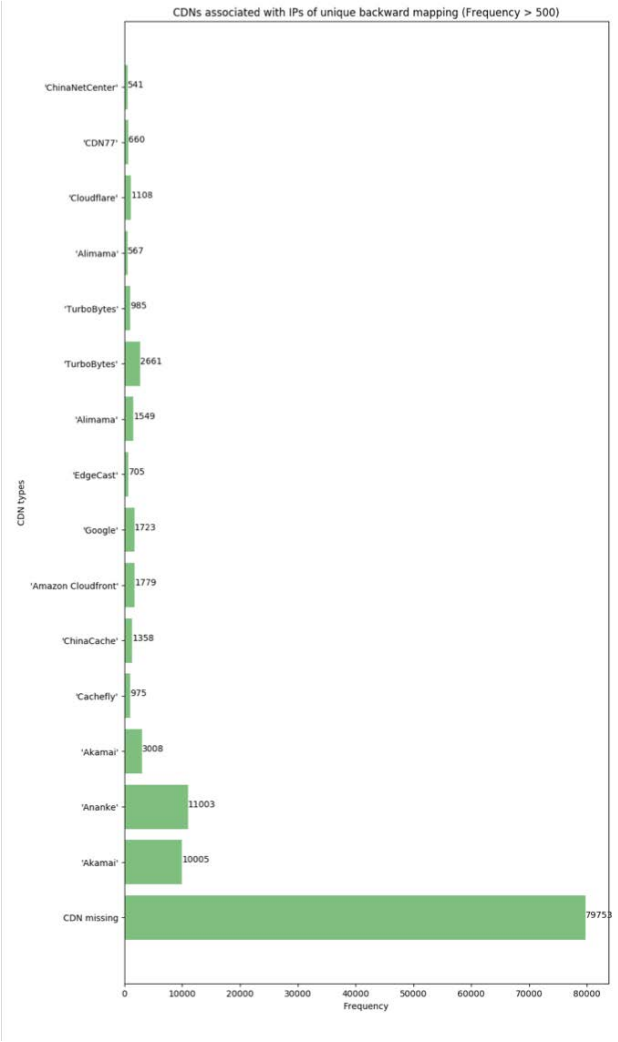

Figure 6: The figure shows the frequency of the IP addresses with a unique mapping to a website that have a CDN associated with them. Approximately 67% of the domains with a one-to-one IP mapping have no CDN associated with them and thus can serve as potential privacy threats as the destination websites can be deciphered by an adversary through a reverse DNS lookup.

However, it is common to assign different customers to different IPs, in part to support older clients that do not send an SNI during an HTTPS connection. As such clients become more rare, consolidating large number of domains to a small number of IPs should be possible.

We also examine the types of resources that are served by "identifying" IPs. Table 2 shows the fraction of resource loads associated with IPs that have an anonymity set of size 1. It is notable that, among popular resource types, fonts are less likely to map to a singleton anonymity set, while XHRs are more likely. However, the size of the anonymity set does not tell the whole story; for example, the IP 104.19.195.151 has an anonymity set size of only 3 domains, but one of them is cdnjs.cloudflare.com. This domain is referenced on over 100 000 sites in our data set, so observing an IP connection to it reveals limited information about what site the user is visiting. On the other hand, table 3 shows requests that map to uniquely identifying IPs, i.e., ones that can be resolved from only a single website in our dataset. About 20% of requests are uniquely identifying in this way; notably, XHRs are less likely to map to site-unique IPs whereas stylesheets and images are more likely. 68% of the IPs in our data set are unique to a single site, and a total of 402 524 (42.6%) of sites use at least one resource whose domain maps to a site-unique IP.

Table 3: Resource types for requests that use unique *IPs*: IPs that are only referenced by a single website. We also show the fraction of all requests of that type that maps to a unique IP.

Request type	Count	% of requests
Document (HTML)	427 501	10.3%
Image	10 140 704	23.3%
Script	3 660 992	15.6%
Stylesheet	2 315 329	26.8%
XHR	355 601	7.6%
Font	426 367	10.5%
Other	391 693	23.8%
Fetch	9 273	2.6%
Media	33 7376	20.1%
EventSource	170	9.9%
Manifest	243	32.4%
TextTrack	34	5.5%
Total	18 065 564	20.0%

3.3 Web Front Pages

We next consider using address fingerprinting on the HTTPS connection to the main server associated with a site. We compute a mapping of the main domain of a site to a set of IP addresses. We then identify the set of websites that map to the same set of IP addresses and compute anonymity sets. The corresponding histogram is shown in fig. 7. 413 576 websites map to a set of IPs that is unique to that site; on the other hand, there is a cluster of 15 520 websites that all map to the same IP address.

3.4 Page Load Fingerprints

In the next experiment, we consider page load fingerprints. We compute a PLF by considering the set of domains that

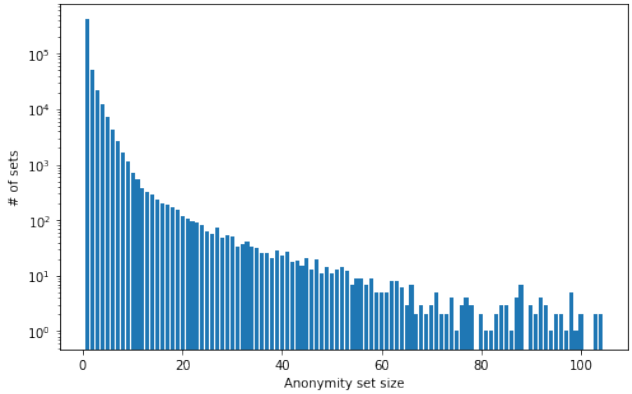

Figure 7: A histogram of IP set anonymity sets corresponding to the "front page" server for each website

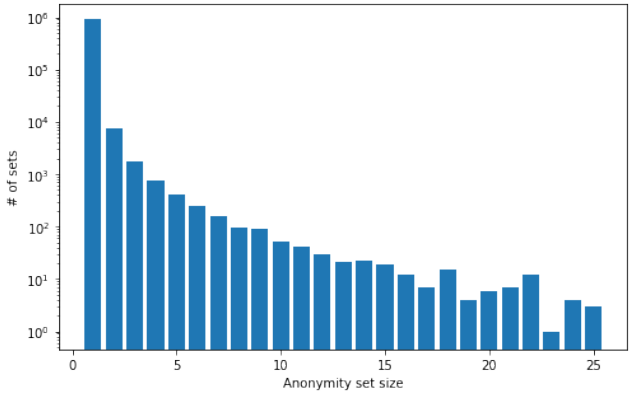

Figure 8: Anonymity set distribution for IP sets. We look up all IPs that correspond to domains that are referenced by any object within a page load. We then count the number of sites that share the same total set of IPs. Among the 944 094 websites in our data set, 903 199 (95.7%) have a unique IP set. On the other hand, the largest anonymity set is a cluster of 903 websites.

are referenced by all of the URLs involved in rendering a particular website and taking the union of the IP addresses that these domains resolve to. When a user visits a website, only a subset of the addresses in the PLF will be contacted, since some domains resolve to multiple IP addresses, only one of which will be chosen, and some resources may be cached at the client. (Additionally, connection coalescing may reduce the number of contacted IPs.) As a result, the fingerprintability of each individual visit to a site may vary. Even if two sites, A and B, have different PLFs, it is possible that a visit to A will produce a set of IPs that matches the PLF for B. However, the difference in PLFs means that it is possible that *some* visit can be mapped to only one of the two sites.

We therefore consider define the the PLF-anonymity set of a website to include only those sites that have an identical PLF. The vast majority of websites (95.7%) have a unique PLF, suggesting that there is a risk of identifying that a user is visiting the site solely from a list of contacted IPs. The distribution of PLF-anonymity set sizes is shown in fig. 8.

4 CONCLUSION

Our measurements show that, in the context of web browsing, DNS and SNI privacy offers limited protection against an adversary who knows a plausible set of sites a user might visit (even if the set is quite large), and who performs forward lookups to infer the domain names and sites associated with given IPs. Using a crawl of Alexa top 1 million sites, we find that nearly half of all IPs involved in the crawl correspond to a unique domain name, and over 95% of sites have a unique set of IPs corresponding to the domains of all the sub-resources. We do identify a significant opportunity for content distribution networks (CDNs) to offer additional protection by coalescing more domains onto the same IP address.

ACKNOWLEDGMENTS

We would like to thank Christopher Wood and Nick Sullivan for helpful discussions about this problem, and Paul Murley for providing us with the data from a million-site scan using MIDA.

REFERENCES

[1] Sara Dickinson, Daniel Kahn Gillmor, and K Tirumaleswar Reddy. 2018. Usage Profiles for DNS over TLS and DNS over DTLS. RFC 8310. https://doi.org/10.17487/RFC8310

[2] Adrienne Porter Felt, Richard Barnes, April King, Chris Palmer, Chris Bentzel, and Parisa Tabriz. 2017. Measuring {HTTPS} Adoption on the Web. In *26th {USENIX} Security Symposium ({USENIX} Security 17)*. 1323–1338.

[3] David Fifield, Chang Lan, Rod Hynes, Percy Wegmann, and Vern Paxson. 2015. Blocking-resistant communication through domain fronting. *Proceedings on Privacy Enhancing Technologies* 2015, 2 (2015), 46–64.

[4] Paweł Foremski, Christian Callegari, and Michele Pagano. 2014. DNS-Class: immediate classification of IP flows using DNS. *International Journal of Network Management* 24, 4 (2014), 272–288.

[5] Google. [n.d.]. Chrome User Experience Report. https://developers.google.com/web/tools/chrome-user-experience-report/.

[6] Paul E. Hoffman and Patrick McManus. 2018. DNS Queries over HTTPS (DoH). RFC 8484. https://doi.org/10.17487/RFC8484

[7] IETF. [n.d.]. Transport Layer Security (TLS) Extensions: Extension Definitions. https://tools.ietf.org/html/rfc6066.

[8] Hongbo Jiang, Andrew W Moore, Zihui Ge, Shudong Jin, and Jia Wang. 2007. Lightweight application classification for network management. In *Proceedings of the 2007 SIGCOMM workshop on Internet network*

management. ACM, 299–304.

[9] B. Laurie, A. Langley, and E. Kasper. 2013. *Certificate Transparency*. RFC 6962. RFC Editor.

[10] Paul Murley. [n.d.]. MIDA: A Tool for Measuring the Web. https://mida.sprai.org/.

[11] Eric Rescorla. 2018. The Transport Layer Security (TLS) Protocol Version 1.3. RFC 8446. https://doi.org/10.17487/RFC8446

[12] Eric Rescorla, Kazuho Oku, Nick Sullivan, and Christopher A. Wood. 2018. *Encrypted Server Name Indication for TLS 1.3*. Internet-Draft draft-ietf-tls-esni-02. Internet Engineering Task Force. https://datatracker.ietf.org/doc/html/draft-ietf-tls-esni-02 Work in Progress.

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

[14] Amazon Web Services. [n.d.]. Alexa Top Sites. https://aws.amazon.com/alexa-top-sites/.

[15] Davide Tammaro, Silvio Valenti, Dario Rossi, and Antonio Pescapé. 2012. Exploiting packet-sampling measurements for traffic characterization and classification. *International Journal of Network Management* 22, 6 (2012), 451–476.

[16] Martino Trevisan, Idilio Drago, Marco Mellia, and Maurizio M Munafo. 2016. Towards web service classification using addresses and DNS. In *2016 International Wireless Communications and Mobile Computing Conference (IWCMC)*. IEEE, 38–43.

[17] Turbobytes. [n.d.]. cdnfinder. https://github.com/turbobytes/cdnfinder.