Knocking on IPs: Identifying HTTPS Websites for Zero-Rated Traffic

Mariano Di Martino
Hasselt University-tUL
Expertise Center for Digital Media
mariano.dimartino@uhasselt.be

Peter Quax
Hasselt University-tUL
Expertise Center for Digital Media
Flanders Make
peter.quax@uhasselt.be

Wim Lamotte
Hasselt University-tUL
Expertise Center for Digital Media
wim.lamotte@uhasselt.be

***Abstract*—Zero-rating is a technique where Internet Service Providers (ISPs) permit consumers to utilize a specific website without charging their internet data plan. Implementing zero-rating requires an accurate website identification method that is also efficient and reliable to be applied on live network traffic. In this paper, we examine existing website identification methods with the objective of applying zero-rating. Furthermore, we demonstrate the ineffectiveness of these methods against modern encryption protocols such as Encrypted SNI and DNS over HTTPS, and therefore show that ISPs are not able to maintain the current zero-rating approaches in the forthcoming future. To address this concern, we present 'Open-Knock', a novel approach that is capable of accurately identifying a zero-rated website, thwarts free-riding attacks and is sustainable on the increasingly encrypted web. In addition, our approach does not require plaintext protocols or pre-processed fingerprints upfront. Finally, our experimental analysis unveils that we are able to convert each IP address to the correct domain name for each website in the Tranco top 6000 websites list with an accuracy of 50.5% and therefore outperform current state-of-the-art approaches.**

I. Introduction

Net neutrality has a prolonged history of debate by both academics and industrial powerhouses. It establishes the question whether Internet Service Providers (ISPs) are legally authorized to discriminate between different types of network traffic that their users consume. Even though the discussion is ongoing, many ISPs have implemented techniques to police network traffic in order to improve performance, decrease financial costs or enhance customer experience [38]. One of such approaches, called 'Zero-Rating' is a technique in which the network traffic generated from the use of certain websites or protocols are not being charged on the consumers data plan and hence permit them to spend a virtually unlimited amount of such Internet traffic. In 2014, zero-rating has gained attraction as it lead to numerous initiatives from both Content Providers (CPs) and ISPs such as Twitter Zero [45], Wikipedia Zero [3] and T-Mobile Binge-On [2]. By using Deep Packet Inspections (DPIs) tools, the adaption of zero-rating had little technical hurdles to overcome, although later on, it raised challenges with regard to identifying the zero-rated website in a cost-effective manner, while still remaining accurate and secure. Implementing new protocols for zero-rating is rarely a viable option, as it would require support from various Internet organizations ranging from browsers to web servers, even though these have been vocal in criticising such approaches [1], [24], [50]. Therefore, zero-rating has mostly been integrated into existing architectures by extracting information from plaintext protocols such that zero-rated HTTPS websites can be identified efficiently from a technical as well as a business point of view [16]. Despite their efficiency, a predominant number of approaches of zero-rating are prone to free-riding attacks in which consumers are able to circumvent zero-rating implementations and as a consequence, are not being charged for normal non zero-rated traffic [34]. On the other hand, cases where a consumer was visiting a zero-rated website while erroneously being charged for that traffic, have also appeared [22]. For this reason, an accurate and secure identification of HTTPS websites is necessary in order to maintain zero-rating functionalities in the forthcoming future.

Similar methods that are being deployed for zero-rating, are also applied by various network actors such as Content Distribution Networks (CDNs) for other network classification problems. Over the last decade, these network actors were able to analyse network traffic from a number of protocols such as plain DNS or plain HTTP to, for instance, gain insights into the websites that their consumers are using [58], to comply in law enforcement cases or to protect minors by using parental control software [12]. However, revelations of privacy abuse over the last years have given rise to an extensive growth of encryption protocols on HTTPS level such as Encrypted Server Name Indication (ESNI) and DNS over HTTPS (DoH). It is clear that those protocols are making it significantly more difficult for providers to efficiently conduct these analysis without active support of the CPs or browsers [18], [42]. Solutions to address this problem have been explored in previous studies of website/webpage fingerprinting, in which a website is identified based on encrypted network traffic samples. However, such methods are not actively being used in practice due to several limitations [23], [33].

Nevertheless, the inability to infer which websites a user is visiting together with the possibilities to abuse current zero-rating implementations on a large scale, will inevitably result into substantial financial losses or shifts in business models for ISPs and DPI vendors. To solve these concerns, we require a method that is able to accurately identify zero-rated websites in encrypted network traffic, while preventing the abuse of free-riding attacks.

In this paper, the following contributions are presented:

- We discuss and summarize existing zero-rating approaches for the identification of HTTPS websites.

In order to demonstrate the inherent limitations of such approaches, free-riding attacks are examined that may result into substantial abuse, as a majority of the considered ISPs are vulnerable to such attacks.

- Furthermore, we propose 'Open-Knock', a novel website identification method specifically designed for zero-rating purposes. Here, we are able to identify a single HTTPS website based on a single IP address without requiring pre-processed fingerprints or preselected websites upfront, rendering our method scalable for live network traffic. Optionally, to increase accuracy, our method is able to apply webpage fingerprinting (WPF) techniques on the encrypted traffic, while solving the limitations present in previous WPF approaches.
- To conclude, we simulate our 'Open-Knock' technique on the Tranco [37] top 6000 websites. In this experimental analysis, we demonstrate that it is possible to identify and therefore, zero-rate 50.5% of the HTTPS websites based on a given IP address. The accuracy can be increased to 56.6% if a network traffic trace of the website visit is available. Hereby, we outperform previous state-of-the-art work in terms of accuracy and applicability. As an additional result, the experiment demonstrates that some properties of encrypting domain names through ESNI and DoH are significantly weakened, depending on the architectural network choices made by these websites.

The dataset of our experiments and an online demo tool is also provided in order to enhance reproducibility and future research on this topic [1].

II. Background

To better understand our analysis of the existing zero-rating implementations and our proposed methods, we first clarify a few concepts in a TLS handshake for HTTPS communication. Starting with Section II-A, we discuss the Server Name Indication (SNI) and the encrypted variant, the ESNI extension. Further, Section II-B describes the use of X.509 certificates in TLS handshakes and finally, Section II-C discusses the main concepts and limitations of WPF techniques.

A. *(Encrypted) Server Name Indication and DNS over HTTPS*

The Server Name Indication (SNI) [28] is an extension for the TLS protocol that specifies to which hostname (e.g. website), a browser or any other client wants to connect. Attaching this extension at the start of a TLS handshake (more specifically, in the Client Hello record) enables the destination webserver to handle multiple websites (and thus, certificates) on the same IP address and only deliver the certificate that the client requests through the SNI. Due to the scarcity of the IPv4 address space and the worldwide partial support of IPv6, it is more beneficial for a server provider to use the SNI as a potential solution and therefore, limit the number of IPv4 addresses it has to buy and maintain. Moreover, it is also not unusual for shared hosting providers to allocate one IP address for all or most of their clientele websites, thereby sharing the same TLS certificate as is often the case with CDNs [21]. The SNI is therefore a widely supported extension on the modern web today.

As the initial RFC draft [28] of this extension was already published in 2003, all modern clients support SNI by sending the domain of the requested website as part of the handshake. Even though TLS<=1.2 provides encryption[2], it only does so at the end of the handshake, hence sending the SNI in plaintext. As a result, any Man in the Middle (MitM) is able to extract the plaintext domain name from the network traffic, which in turn, is utilized for a wide range of industrial and academic work as we will show in Section III-A.

Because privacy advocates have found the SNI extension to be problematic for the consumers privacy, a new extension called 'ESNI' [47] was introduced in 2018 that encrypts the original SNI by a derived symmetric key from, for instance, previous DNS communications. More specifically, the client requests the ESNI record of a given domain name from a ESNI-enabled DNS resolver. A Diffie-Hellman exchange is then setup to derive a symmetric key with the DNS resolver, which can then be used to encrypt the original SNI and sent as part of the ClientHello. A hash is also included into the ESNI extension that ties the public keyshare of the ClientHello to the extension itself. To this end, the ESNI cannot be used with another ClientHello, preventing a MitM to replay the same ESNI with his own ClientHello. Clearly, both DNS resolver and webserver must support ESNI to avoid a potential fallback to standard SNI. There are however additional requirements to completely adhere to ESNI guarantees of protecting the domain name: *(i)* TLS 1.3 encrypts the ServerCertificate record – a design choice not implemented in TLS 1.2. This is important as the ESNI suggests which certificate to send, meaning that a plaintext certificate may indirectly leak the domain name. *(ii)* The ESNI record must be requested through an authenticated and encrypted channel (e.g. DNS over HTTPS) between client and DNS resolver as to prevent a MitM to extract the record. As of June 2020, Firefox and Chrome have implemented ESNI and several organizations such as Cloudflare and Google have also made ESNI-enabled DNS resolvers available for public use [4], [46]. Note that a recent update to the ESNI draft [47] has introduced some technical changes related to encrypting the completed Client Hello record and has also renamed ESNI to *ECHO* [48].

Finally, DoH [30] is a protocol that allows clients to securely communicate DNS request and responses over HTTPS by using an intermediate DNS resolver. Compared to plaintext DNS over port 53, this protocol usually operates over port 443 and encrypts all DNS communication. Since June 2019, DoH is supported by Mozilla Firefox and Google Chrome. Despite DoH solving several inherent problems with plaintext DNS, opinions are divided whether centralizing all DNS communication to a single intermediate resolver by default will enhance the overall privacy of end consumers [55].

B. *X.509 Subject Alternative Names in TLS*

Before a client can communicate with a webserver over HTTPS, a TLS handshake must first be performed. Further

[1] https://open-knock.com/openknock.php

[2] TLS 1.3 encrypts records after the ServerHello, so the SNI is still sent in plaintext.

down the handshake, the ServerCertificate record usually provides the client with a X.509 public key certificate [19] that proves the identity of the webserver. The choice of which certificate to provide the client, is typically based on either the IP address to which the clients request was made or on the SNI as we mentioned in Section II-A. The name of the identity is contained in the *Common Name* field of the certificate, which the client has to verify whether it matches the domain name of the website that it wants to access. However, in case the certificate is issued for multiple domain names, the Common Name field cannot accomodate all domains and thus, an additional field in the certificate called Subject Alternative Name (SAN) is introduced. This field contains all the domain names that the certificate controls and makes multiple websites behind one IP address much easier to manage.

Moreover, the SAN field is even necessary for older clients that do not support SNI as they do not offer a method to indicate the domain name during the TLS handshake. A non-SNI supporting client is then able to freely pick which domain name it wants to access from the SANs. Although, the SNI extension is widely supported by all modern browsers [3], it is often possible to act as a non-SNI client and hence receive a list of SANs which indicate some or all domain names that the IP address or underlying webserver may handle. Prior work that performs such extraction exists [31], but have not yet been explored in a zero-rating context.

C. Website fingerprinting

When communicating over HTTPS, sensitive data such as URLs and HTTP body data is encrypted and thus protected against adversarial MitMs. However, metadata is usually not considered to be sensitive as the timing and sizes of HTTP packets in TLS records are still determinable by an adversary. By exploiting the characteristics of precisely such metadata, a MitM is able to distinguish between different webpages or websites as it can observe patterns in the communication between client and server. The detection of these patterns lays the foundation of webpage or website fingerprinting (WPF) in general. Cheng and Avnur [14] were the first to consider the possibility of fingerprinting webpages over HTTPS. Over the last decade, a considerable number of works have build upon this initial proof of concept and have devised new techniques to further improve practicality and accuracy [8], [41], [43]. An extensive overview of WPF attacks over HTTPS is presented in previous work and is therefore out of scope in this paper.

Instead, we discuss the general considerations necessary to understand the concepts applied in our paper. An in-depth study related to these concerns is presented by Juarez et al. [33]. First, *(i)* prior work is primarily focused on fingerprinting the homepage of each website, instead of collecting multiple webpages per website. As a result, such a WPF classifier is not able to accurately predict other webpages contained in that same website. In addition, *(ii)* a collection of webpage fingerprints has to be constructed before being able to predict the specific webpage being visited. For instance, if an adversary is interested in predicting all webpages from Wikipedia, it first has to multisample each and every webpage on Wikipedia in order to construct the necessary fingerprints, which is in most cases an impractical task. And finally, *(iii)* the variety of modern clients is significant, leading to different patterns in network traffic and thus causing WPF attacks to substantially decrease in accuracy when performed on clients (e.g. browsers) not seen yet.

Furthermore, the objective of such studies is often not to provide zero-rating functionality, but to reveal the possibilities of bypassing certain protocols or countermeasures such as Tor or VPNs. Therefore, realistic models for WPF are known to have an accuracy that is far lower then is acceptable for zero-rating. Indeed, for zero-rating, it is evident that the identification of a given website or webpage should have an exceptionally low number of false negatives as to prevent concerns of consumers using a zero-rating website, while still being charged on their data plan due to a potential misclassification.

III. Related Work

Tracking and analyzing behavioural patterns of users through the use of plaintext domains such as DNS and SNI have been explored extensively in previous work [29], [35], [36]. As DoH and ESNI have been introduced in 2018, little work has been conducted from a security and privacy angle. Patil et al. [44] have performed an internet measurement experiment where the top 1M Alexa websites are crawled and the uniqueness of each resulting IP address is analyzed. In their work, they demonstrate that a single domain name can be inferred from 48% of the websites in a closed world scenario[4]. The measurements are produced from a set of 1M websites, which means that, if one wants to maintain the desired accuracy, they should regularly restart the crawling process to update the resulting set of IP addresses. Furthermore, Siby et al. [52] have proposed a n-gram model to predict the domain names from DoH based on the sizes and timing of the packets. They achieve a F1-score of 70% in an open world scenario of 1500 monitored webpages and demonstrate the robustness against countermeasures such as padding. Here, they acknowledge that their resulting score is sensitive to the location of the client and DNS resolver and therefore decreases significantly when performed in a different setup.

A. Website identification methods

Although website identification methods specifically for zero-rating have been explored in academic literature, not many have studied the actual implementations of zero-rating as carried out by ISPs and CPs [34], [38]. Numerous patents that incorporate zero-rating have been filled by DPI vendors, ISPs and CPs such as Facebook [27], T-Mobile [6] and Cisco [57]. Despite the significant financial effort that have been put into these techniques, we show nevertheless that they have the same technical limitations and vulnerabilities that we observe in academic literature. One of these limitations is the inability to automatically detect or thwart free-riding attacks, in which an attacker can modify his own network traffic such that it resembles a zero-rated service or website. Another concern is that most of the current approaches are unsustainable in the near future as they do not support the rise of new encryption

[3] Exceptions: e.g. IE 6.0 pre-Windows Vista and Android 1.X-2.X.

[4] A scenario where only a number of pre-selected or pre-processed websites can be identified.

protocols, which we argue will eventually eradicate many of the existing methods.

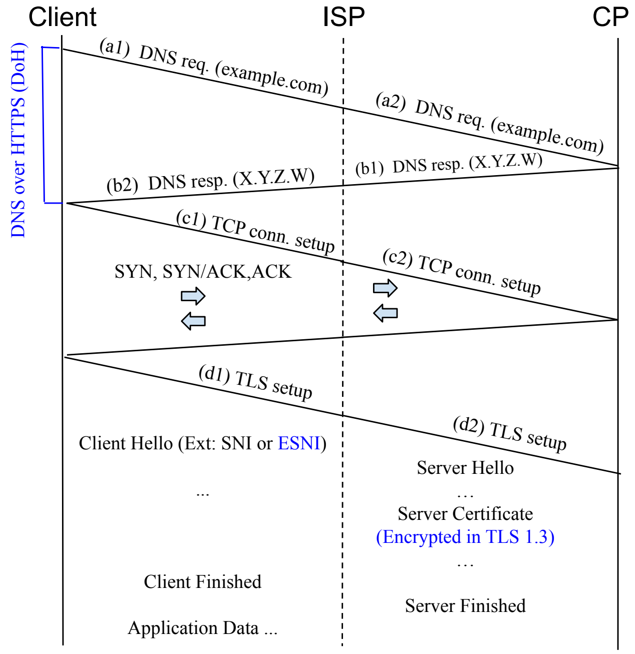
In this section, we discuss these website identification methods in more detail and analyze whether they are suitable for detecting zero-rated traffic. Furthermore, we explore the practical considerations, the possibilities of free-riding attacks and finally, examine the long-term properties of these methods. To better explain these approaches, a flow of an average consumer browsing the web is depicted in Figure 1 and we will refer to this figure throughout this section. A summary of the existing website identification methods is provided in Table I and the columns of this table are explained below:

- *HTTPS*: The method is able to operate with HTTP over TLS traffic.
- *Support*: The method requires support from the destination webserver or client (e.g. installing certain software on the client).
- *In-band*: The method does not require active out-of-band communication to operate correctly. E.g. a passive MitM can be used that only reads network traffic but does not modify, drop or append additional traffic.
- *ESNI and DoH*: The method works on network traffic with the ESNI and DoH protocol present.
- *Free-Riding I*: The method is robust against free-riding attacks where an additional MitM is deployed (for instance: a webproxy).
- *Free-Riding II*: The method is robust against free-riding attacks without the need of additional MitM network nodes.
- *Open set*: The method requires no preliminary knowledge about the possible websites that can be identified.

Our proposed method 'Open-Knock' is also listed in the table and discussed in Section IV.

1) Non-HTTPS: Identifying websites in non-HTTPS traffic is easy to accomplish, as it is unencrypted. Kakhki et al. [34] showed that some ISPs such as T-Mobile are parsing HTTP headers in order to identify the website. For instance, extracting the 'Host' header may indicate the website being used, while the 'Content-Type' header specifies the type of network traffic. By parsing these headers for each HTTP request and response, the ISP is able to accurately zero-rate a specific type of traffic from a specific website. However, as it is possible for a client to modify these headers, this approach is trivial to circumvent. As of June 2020, 85% of the top 1 million websites and 85% of the page loads on the internet are using HTTPS by default [5], [54], thereby making it impossible to parse plaintext HTTP headers for the majority of these websites.

Fig. 1. Webflow of a client that browses to a website 'example.com' over HTTPS, while the ISP is a MitM. First, a DNS request is sent and the corresponding DNS response is received. Next, a TCP connection to an IP in the DNS response is initiated. Furthermore, a TLS handshake is performed and finally, the actual HTTP data is sent in Application Data records.



2) Whitelisting IPs: Whitelisting IP addresses that correspond to a given website is one of the earliest approaches for zero-rating HTTPS traffic [3]. Here, a CP provides a list of IP addresses to the ISP, where each IP address is somehow used by the zero-rated website of the CP. An ISP is therefore able to identify all packets that are destined to or originating from these IP addresses by simply deploying a form of IP matching rule (may be performed at *c1-c2* in Figure 1). Although this method is difficult to circumvent, it does however require cooperation of the CP as it is often impractical to gather a list of all possible IP addresses that the zero-rated website may use. A CP usually does not have control over the IP addresses from providers that are hosted externally (e.g. CDNs or web analytic services) as they may introduce new IP ranges without prior notification. Organizations such as the Wikipedia Foundation publicly provide a list of IP ranges, which can be utilized to zero-rate their Wikipedia network traffic [3] and it has been deployed by several ISPs in South-Asia and regions of Africa.

3) SNI: Since the SNI extension of the TLS protocol is sent in plaintext, it is an efficient method to identify webservices in network traffic as demonstrated by several works [26], [34], [39]. More specifically, Kakhki et al. [34] show that T-Mobile's BingeOn mobile plan extracts the SNI hostname and compares this to predefined domain names with simple string matching such that video streaming services from various CPs can be zero-rated. Similarly, Li et al. [38] discovered that in some proprietary middleboxes, a simple string matching is performed on the first two TLS records without parsing the SNI, presumably to increase DPI efficiency. A large number of commercial DPI vendors [5] have implemented this technique and we can confirm that 2 large ISPs in Europe are currently deploying this technique for zero-rating. In another instance, the Qosmos engine publicly describes their website identification method using SNI[6]. They also acknowledge that some old clients do

[5] https://www.paloaltonetworks.com/network-security
https://www.baracuda.com
https://www.clavister.com

[6] https://www.qosmos.com/blog_qosmos/yes-encrypted-traffic-can-be-classified/

TABLE I. SUMMARY OF WEBSITE IDENTIFICATION METHODS FOR ZERO-RATING. '≈' INDICATES A DEPENDENCY ON THE SPECIFIC ZERO-RATING IMPLEMENTATION USED.

Method	HTTPS	Support	In-band	ESNI and DoH	Free-Riding I	Free-Riding II	Open set	Known usages
Non-HTTPS			✓			✓	✓	Various ISPs [16], [34]
SNI	✓	Post−2006 clients	✓			≈	✓	Various ISPs, censorship [13]
Whitelisting IPs	✓	CP	✓	✓	✓	✓		Wikipedia Zero [3], censorship [13]
DNS	✓		✓			✓	✓	Facebook Zero [27], censorship [13]
Verify SNI by DNS	✓					✓	✓	Cisco patent [57]
Network Cookies	✓	CP	✓	✓	?	?	?	?
WPF	✓		✓	✓		≈		Censorship [25]
Self-signed	✓	Client	✓	✓		✓	✓	Enterprise and surveillance governments [17], [20]
Reverse DNS	✓	CP		✓	≈	✓	≈	Anti-spoofing, OSINT tools (e.g. Maltego)
Open-Knock	✓			✓	✓	≈	✓	

not support SNI and therefore, have incorporated the ability to extract the domain name from the certificate (ServerCertificate record) later in the TLS handshake. However, the recent advancements of TLS 1.3 and ESNI will result into TLS handshakes where the SNI extension and certificates are encrypted, rendering such approaches infeasible [13]. Finally, two free-riding attacks exist that are able to circumvent this method: *(i)* Some web servers ignore the actual value in the SNI [51], thereby making it possible to setup a zero-rated connection directly by modifying the SNI in the Client Hello record to a domain name that is known to be zero-rated (may be performed at *d1* in Figure 1). Or *(ii)* an external webproxy is setup that will ignore the modified SNI and re-route all traffic from the client to the destination and back.

4) DNS: In a common scenario where a client wants to browse a website, a DNS request is first sent to translate the domain name of the website to an IP address. As described in patents of Facebook Inc. [27] and Cisco [57], an ISP is able to intercept the DNS response to that request and examine whether the domain name should be zero-rated (performed at *b1-b2* in Figure 1). If that is the case, the corresponding IP address is appended to a whitelist and can further be used to zero-rate the corresponding traffic, as discussed in Section III-A2. However, the aforementioned approach is also susceptible to free-riding attacks if the client is allowed to choose his own DNS server. For example, the client can set up a DNS server that always responds to a DNS request with the IP address of a zero-rated website regardless of the domain name being requested. Even if the ISP would force their consumers to use an ISP deployed DNS server, caching would then be another point of concern. A smartphone device may cache zero-rated IP addresses of a previous DNS response from ISP A (e.g. mobile data provider) and then switch his smartphone setting to ISP B (e.g. Wi-Fi provider), resulting into no additional DNS requests that can be examined and thus zero-rated by ISP B. Furthermore, protocols that encrypt DNS communication such as DoH are growing and may interfere with the interception of those DNS requests and responses.

5) Network cookies: Yiakoumis et al. [59] present a general framework for identifying types of network traffic, by using network cookies. A network cookie is an HTTP cookie that contains information related to the type of traffic (e.g. domain name of the website). Here, ISPs or CPs provide a network cookie to the client and the client attaches that cookie to the HTTP request or response that needs policing. As the network cookie is signed, alteration is not possible and the ISP is therefore able to extract and verify the cookie from the TCP connection and finally, police (in our case, zero-rate) the network traffic accordingly. However, the authors do not elucidate on the limitations of identifying whether the requested network cookie is actually attached to the traffic that it was meant for. Although they suggest to add additional attributes to the cookie that indicates the traffic to which it belongs, it does not demonstrate how such indication should be made. In other words, the validity of the request for a network cookie can only be verified with cooperation of the CP. As a result, the CP has to fallback to other means of verification; for instance, the client may include the IP address in the signed cookie to which it wants to connect and the CP then has to verify whether the information is correct.

6) Reverse DNS: A reverse DNS lookup is an out-of-band method to resolve an IP address to a domain name by utilizing DNS PTR records. For instance an ISP may request (as soon as *a1* is captured in Figure 1) the DNS PTR record for IP address 'X.Y.Z.W', which will return the corresponding domain name 'example.com'. According to RFC 1912 [10], it is recommended that each IP address should be resolvable to a hostname by using PTR records, although in practice, this is often not the case as CPs have little incentive to do so [11]. Even in cases where a hostname is returned, it frequently does not specify the original domain name but instead, indicates the hosting or cloud provider of the server. For instance, the IP address '54.230.187.137' which currently serves the domain name 'reuters.com', returns 'server-54-230-187-137.cdg3.r.cloudfront.net' as a hostname in the PTR record. If a reverse DNS lookup of an IP x returns domain y while a forward DNS lookup of domain y returns IP x, then we call this 'forward-confirmed DNS'. Forward-confirmed DNS is utilized in a wide variety of techniques, such as anti-spam email and Open-Source Intelligence (OSINT) tools[7]. Although it may be used as an additional method for zero-rating, it is not possible to accurately identify domain names with reverse DNS lookups alone due to the low number of usages [44]. Finally, the DNS resolver should be trusted by the ISP, as a consumer may setup a fake DNS resolver similar to the free-riding attack discussed in Section III-A4.

[7] https://www.shodan.io
https://www.paterva.com/buy/maltego-clients/maltego-ce.php
https://reverseip.domaintools.com/search/

7) Self-signed certificates: The cornerstone of why the majority of the HTTPS connections are encrypted and authenticated and therefore not readable and modifiable by a MitM, is due to the use of certificates. Certificates are issued by a Certificate Authority (CA) who is an external party trusted by all modern clients. If a client receives a certificate from a server in a TLS handshake (Server Certificate record in Figure 1), it must first verify with the CA whether the certificate is valid and provided for a specific domain name. Further traffic is then encrypted with, i.a. the server certificate, and can only be read by a party that has access to the private key of the certificate (e.g. the webserver). However, if a client does not verify the certificate, any MitM is able to construct a similar certificate and decrypt further communication with his own private key between client and server. For security software such as antiviruses and Intrusion Detection System (IDS) in enterprise environments, it is common practice to issue self-signed certificates as they are then able to inspect the web traffic and identify possibly malicious websites [20]. Accepting a self-signed certificate is implemented by making internal changes to the client or operating system of the client such that it will trust a given CA. An ISP could, in theory, also issue self-signed certificates such that it can decrypt the traffic and identify each website (similar as it would do with non-HTTPS traffic). Nevertheless, it is clear that such approach is impractical in the long term and would negatively affect the overall privacy of the end users. In 2019, the government of Kazakhstan has issued a notice for citizens to install self-signed certificates [17].

8) Verify SNI by DNS: Shbair et al. [51] propose to extract the SNI value from each TLS handshake (Client Hello in Figure 1) and subsequently, verify the authenticity of the value by performing an out-of-band DNS request to a trustworthy DNS server with as input the extracted SNI/domain name. Only if the destination IP address of the TLS handshake (from the initial TCP connection) is included in the DNS response, the authenticity of the SNI can be proven and thus the website identified. By introducing this verification step, it is not possible to alter the SNI as the IP addresses in the subsequent DNS response will not match, thereby mitigating this type of free-riding attack. Clearly, such approach postulates that a SNI value is always provided in each TLS handshake. As we have pointed out before, this is often not the case with modern protocols such as ESNI and with older operating systems that do not support SNI. Moreover, the authors also observe that the IP addresses in the DNS response may not always include the initial IP address due to DNS load delegation and balancing. Even though, their experiment shows that the IP address is included in 83.75% of the cases, our experimental analysis of Open-Knock have found that this number has decreased significantly over the last years (Section IV-G). A very similar patent that performs this identification method has been filled by Cisco Inc. [57] in which a proxy device is used to perform the verification.

9) Webpage fingerprinting: In WPF, each website or webpage consists of a unique fingerprint based on network traffic samples or specific properties of the website/webpage. By collecting, for instance, the timing and size of TCP packets when browsing the website, fingerprints of these websites can be constructed by various mathematical techniques such as neural networks or other machine learning models. The WPF model in question will then predict to which particular website, a new (not-seen) network trace belongs to. Although numerous studies have proposed WPF models to identify websites based on HTTPS traffic traces [8], [43], they often fail to perform well in realistic scenarios due to the variability of browsers, protocols and intermediate network nodes [9], [23], [33]. Moreover, as these models are inherently vulnerable to statistical changes, it is possible for an attacker to circumvent these by using adversarial samples [32] or by applying WPF defenses to web clients or servers [7], [15]. As discussed in Section II-C, the accuracy of realistic models for WPF are usually not stable enough to be utilized for zero-rating. With the exception of the ‘Great Firewall of China’ [8], we are not aware of any ISP or DPI vendor that has incorporated webpage fingerprinting for zero-rated traffic in one of their products.

IV. Open-Knock: a Website Identification Technique for Zero-Rating

A. Context

Before illustrating the details of our method, we first lay down a practical scenario in which our method is meant to be utilized. Our environment consists of the same parties as observed in Figure 1. The ISP, who is currently acting as a MitM, would like to identify whether their consumers are using one of the zero-rated websites such that the corresponding zero-rating policy can be applied correctly. In order to predict the domain name of that website, the ISP can extract the IP address of all TCP connections of the consumer network traffic. By providing these IP addresses to our method, a resulting list is outputted that matches each IP address with the domain name of the website that was visited by that particular consumer.

Accordingly, we present the details of our novel approach in identifying HTTPS websites for zero-rating, called ‘Open-Knock’. Here, the domain name of an HTTPS website is identified based on an initial IP address I, captured from a network trace. In addition, our process also consists of an (optional) step that applies webpage fingerprinting in which the complete metadata T from all TCP connections to that IP address I are required. By deploying Open-Knock, we are able to evade some protective properties of DoH and ESNI and are able to extend the practicality of prior works that are based on the extraction of plaintext domain names such as DNS and SNI. A summary of Open-Knock is presented below:

1) Extract domain names from SANs and perform reverse DNS, given IP address I.
2) Filter all domain names that are not reachable, fail certificate checks or do not accept an HTTPS request.
3) Expand the list by crawling existing domain names from the corresponding website, to find new reachable domain names.
4) Exploit DNS delegation characteristics to filter out domain names that are not handled by IP address I.
5) Predict the possible domain name(s).
6) Optional: If multiple domain names are left, then fingerprint the remaining domain names in real-time

[8] A Chinese government-owned surveillance framework to regulate all domestic internet traffic [25].

(applying metadata of T) and create a WPF model to produce the final prediction.

Each chronological step in the process is represented by a subsection. The output list of domain names in each step is the input list of the next step, of which each output list is called R. At the end of the process, list R will contain the possible domain names that are matched to the initial IP address I and (optional) network trace T. Given an IP address I, OpenKnock can either produce a prediction with a relatively high certainty or it can produce no prediction at all. In the context of zero-rating, we argue that producing no prediction at all is substantially better than producing a prediction with a low certainty as an ISP usually already has a substantial amount of network traffic available and therefore, prefers accurate and qualitative output data over quantitative output data for applying their zero-rating policy.

B. *Extract domain names*

We setup a DPI-like MitM that monitors traffic flowing from the client to the destination webserver. For each targeted TCP connection that the client initiates to an unknown IP address on port 443, the IP address I is saved by the DPI and used as an input element to identify the corresponding domain name.

The first step in the process starts when a new IP address is saved. Here, the MitM initiates 4 new out-of-band TCP connections to that IP address on port 443 and performs a TLS 1.1 handshake for each of those. This handshake is initiated with the following 4 options:

1. Perform a TLS handshake with valid parameters and a valid SNI extension, as any modern browser implementation would do.
2. Perform a TLS handshake with a range of pre-2006 cipher suites and add a SNI extension with the value set to 'example.com'. This simulates a client that does support SNI, but not any of the ciphers that were introduced after the SNI.
3. Perform a TLS handshake with a range of pre-2006 cipher suites and add a SNI extension with an empty value and set the length to 0. Although, an empty SNI is allowed by RFC6066 [28], many implementations show odd behaviour when encountering this.
4. Perform a TLS handshake with a range of pre-2006 cipher suites without a SNI extension. This simulates the behaviour of an old client that does not support SNI.

Depending on the server implementation behind the IP address, each separate option may return different domain names in the SANs field of the TLS Server Certificate record. Our analysis shows that option 1 and 4 combined, will return approximately 96% of the domain names. All extracted SAN domain names of the chosen options are collected and used as the initial input list R for the next step in the process. In addition to the domain names gathered from the SANs, we also request the PTR record through reverse DNS from IP I and add the received domain name to the initial input list R. To reduce the risks of free-riding attacks, it is necessary to deploy an additional step to thwart possible circumventions of a webserver deceitfully acting like it owns a specific domain name. This step consists of verifying whether the received certificates are signed by a trusted CA and that the destination server has access to the private key of that certificate. In order to do so, the TCP out-of-band connections complete the TLS handshake for every domain and verifies the correctness of the certificate.

C. *Filter unreachable domains*

With the input list R of domain names, we recursively request the A-record of each root domain name[9] through DNS . Each returned A-record is examined and the corresponding domain name in R is subsequently removed from the list, i.f.f. the A-record is empty such that we filter out all domain names that are not reachable externally. Next, from the IP addresses in those remaining A-records, we take the first IP of every domain and perform an HTTPS request on port 443 with the SNI set to that domain name. If the request fails or if it does not return a status code between 200 and 400, then we also remove the complete A-record and the corresponding domain name from the list R. To this end, all domain names that are impossible to be reached through this IP address are removed. Finally, possible HTTP redirects from the remaining domain names are captured and the original domain name in the list will then be replaced by the root domain name of the redirect.

D. *Expand by crawling domains*

In this step of the process, the HTTPS websites behind the remaining domain names may contain links to other websites of a different domain Y, yet the server behind IP address I is also able to provide a X.509 certificate for Y. For instance, the homepage of a premium shared hosting provider often contains URLs to the websites developed for their clients, of which some of the client websites accept SNI values of other client websites due to a shared certificate or CDN-type server behind the hosting providers IP address. To take advantage of this information, the homepage of each remaining website in R is crawled and all direct URLs pointing towards other domain names are captured. The captured URLs are then visited again through the initial IP address I by performing a complete HTTPS request with the SNI set to the domain name from the captured URL, as to verify if they are reachable and a valid certificate is available. Here, the domain name is reachable if the corresponding HTTPS request does not fail and the received HTTP status code is between 200 and 400. Root domain names of reachable URLs are extracted and added to the list R.
For example: we start with a hosting provider that owns website A and B, but only A is in R. Crawling through the homepage of website A, returns the domain names B and C. For both domains, we initiate a TCP connection to IP I, set the SNI in Client Hello to the corresponding domain name of the website (A and B) and finally, request the root webpage of the website. Only the domain names of the requests that have succeeded will be added to R.

[9] for instance, the domain name 'help.example.com' will result into the root domain 'example.com'

E. Exploit DNS load balancing and delegation

Despite the fact that the destination webserver can provide numerous SANs in one shared certificate or that it chooses one based on the IP address, it does not necessarily mean that all DNS requests to these domain names lead to the same set of IP addresses. For instance, domain X and domain Y share the same X.509 certificate provided by a TLS handshake to IP address I whereas a DNS A-record request to domain X and domain Y may both return a different set of IP addresses, regardless if address I is included. In general, we observe that there are 2 scenarios in which such behaviour can occur: *(i)* the DNS load balancer is distributing the workload of the destination webservers by shifting the set of returned IP addresses in a round-robin or parameterized fashion. *(ii)* the IP addresses are pre-allocated to certain domain names and thus the DNS resolver simply returns a mostly fixed set of IP addresses for each domain, regardless of how the corresponding TLS certificate was constructed.

The inner-workings of scenario (i) are difficult to predict as the 'balancing' may depend on several parameters such as the server computing capacity or location. However scenario (ii) has, to the contrary, a relatively predictable outcome. To examine if a CP has implemented scenario (ii), we conduct the following steps for each remaining domain name d in R:

- First, we issue a recursive DNS request for domain d and save the IP addresses of the response in $P_0(d)$.
- Introduce a delay of q seconds (q=10, by default) and then re-issue the same DNS request for domain d and save the IP addresses in $P_q(d)$.
- Lastly, numerically order the sets $P_0(d)$ and $P_q(d)$ and compare both ordered sets; if the ordered sets are equal, then we say that domain d is *matched*.

Now, we consider the CP IP addresses to be *pre-allocated* i.f.f. more than half of the domains are *matched*, as small inconsistencies in the returned sets of IP addresses are possible. For example, a CP may perform additional load balancing by removing IP addresses from the DNS records that are not accessible at that time, or by only showing a random subset of the IP addresses allocated to that domain name. In the instance that the CP IPs *are* pre-allocated, we will remove every matched domain d from R i.f.f. IP address I is not included in $P_0(d)$ or $P_q(d)$. In case the IP addresses are not pre-allocated, we continue with the same list R as was delivered by the previous step.

F. Identify fingerprint

By now, list R may either have one domain name left (or multiple domain names with the same root name) or multiple domain names with different root names. In the former case, Open-Knock will predict that domain name from R and the process ends. In the latter case, we are unable to reduce the list further and are therefore, unable to predict a single domain name. However, fingerprinting the remaining domain names to make a final prediction is possible but may not achieve the desired accuracy to be deployed in a zero-rating environment. For instance, if the corresponding website of the domain name contains many pages, it becomes difficult to fingerprint the right number of webpages. However, we will discuss the fingerprinting process for all remaining domains for zero-rating policies where such fingerprinting method is desired.

In this fingerprinting process, we consider each domain to be a functional website and will now require a traffic trace T (only containing the TCP connections to, and from IP I) of the client during the visit of the website in question. Strengthened with this additional information, we will fingerprint the remaining websites in real-time, train an existing WPF model and finally, apply our traffic trace T to this model in order to produce a final prediction. First, we have to capture a solid view of each website to construct an accurate fingerprint. In prior works, the homepage of each website is often considered to give a complete view of the website – an assumption that is inaccurate [23]. Therefore, our approach captures multiple webpages over multiple browsers in an attempt to create a more robust fingerprint that works in practical environments. To create our list of possible webpages for each website, we first crawl the homepage of the website with a fixed crawling depth and save each URL that links to the same domain d and has HTML content-type *'text\html'*. A number of h random URLs are then picked. If less than h URLs are available, then pick all URLs. Choosing a large h will generate a more robust fingerprint of the website, but increases network bandwidth and training time. Following the list of URLs, we sample each webpage by conducting the following process with 2 different browsers:

1) We visit each picked URL through a browser by using Selenium[10] and capture all TCP connections that are initiated to IP address I.
2) For each captured TCP connection, the size of each TLS Application Data record from *server to client* is stored in the sample.
3) We chronologically order each size element of the sample, based on the timestamp of the corresponding TLS record.

In our study, we used a desktop version of the latest (since June 2020) Firefox and Chrome browser. Thus in total, a number of $2h$ samples are captured for each website. All samples that are captured in this process are considered to be *training data* for our WPF model. Furthermore, we input our training data into the WPF model presented by Liberatore and Levine [40] and normalize it such as proposed by [56]. This model uses a Naive Bayes classifier where each class represents a website. Finally, we process our network trace T similar as we did with our training data and provide the processed sample T to the WPF model and use its outputted class as our final prediction. Although we could have chosen any WPF model for this step, our preliminary observations show that this model did not require a large number of samples to identify a website accurately as is often the case with more recent models [49], [53] and the currently used model is specifically devised to train only on TLS record size features, such that patterns based on timing differences are eliminated.

G. Results

We conduct an experiment on the Tranco top 6000 website list, where we apply our Open-Knock method to an IP address

[10] https://www.seleniumhq.org/

that corresponds to each Tranco website. In order to do so, we first perform a DNS request for each website (domain name) on the Tranco list and use the first IP address in the response as our initial input for our method. Similar to prior work, all websites in the list that have no homepage on the root path, have invalid TLS certificates, are not available in Europe[11] or take longer than 20 seconds to load, are removed from the list, resulting into 3474 websites. Examples are websites not supporting HTTPS or have blocked our repeated web requests by some type of firewall. Although circumvention of some of these issues is possible, we did not attempt to do so due to ethical concerns. We classify each IP to domain prediction as either *correct*, *incorrect* or *unknown*. A compartmentalization of our results is shown in Figure 2 and 3. In this chart, we observe that 50.5% of all the websites can be identified solely based on the IP address. Moreover, for 9.2% of the IP addresses, an incorrect prediction was produced, often due to webhosting providers (for instance, Wordpress). In fact, many of these providers only support SNI-capable browsers and will therefore redirect all requests without SNI to the default homepage of the provider. Another example of such issue is that some websites have multiple certificates with partially the same SANs, which will also be load balanced over different IPs, making it difficult for our approach to make an accurate prediction. For the remaining 40.3% of the websites (*unknown* predictions), our method could not conclusively decide which domain name to pick and hence, produce no prediction at all. The reasons why Open-Knock could not produce a prediction in these cases are explained below:

Fig. 2. Results of our Open-Knock method applied on the Tranco top 6000, resulting into a total of 3474 websites. *Unknown (in)* are all predictions where the correct domain name is still in the remaining R, while *Unknown (out)* is the exact opposite. *Unknown (empty)* are all the predictions where the resulting R is empty.

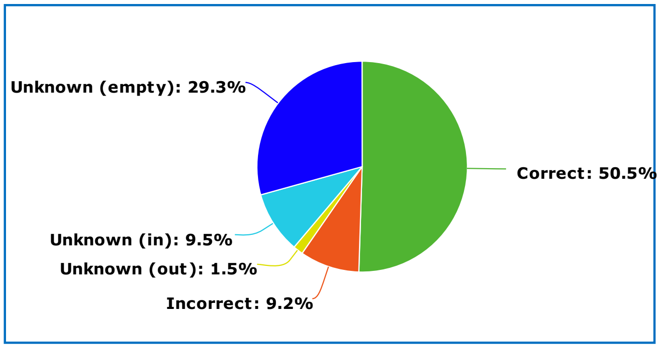
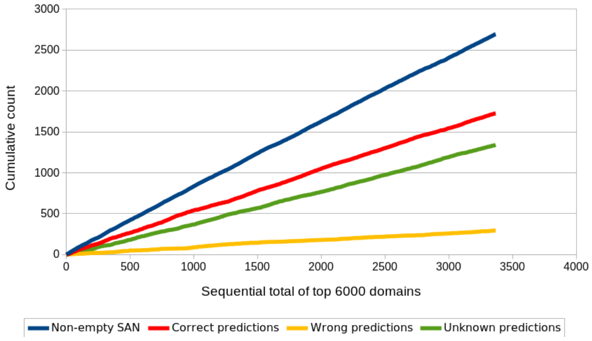


Fig. 3. Cumulative results of our Open-Knock method applied on the Tranco top 6000.



- *Empty SANs (19.4%)*: When extracting the domain names from the SANs (Section IV-B), there were no domain names returned. Those websites are likely not supporting non-SNI capable browsers. As a result, no further steps can be performed.
- *Incorrectly filtered SANs (1.5%)*: The correct domain was returned from the SANs, but our subsequent steps incorrectly filtered out that correct domain name.
- *Correctly filtered SAN (10.9%)*: One or more domain names were returned from the SANs, but none of these indicated the correct domain name. Our method filtered out these domain names, resulting into an empty R.
- *Multiple SANs left (8.5%)*: One of the SANs contained the correct domain name, but Open-Knock was not able to filter all the other remaining domain names.

Interestingly, only 3.4% of the IP addresses in the top 6000 return a PTR record that is equal to the corresponding domain name, compared to 1.9% in [44]. For all the *unknown* predictions which have 2 or more domain names in R (11.0%), we apply a website fingerprinting method (Section IV-F) on the websites of each corresponding domain name. However, note that it may be possible that the correct domain name is incorrectly filtered out of R in another step of the process, meaning that the WPF model may never produce the correct prediction [12]. Although the relative accuracy of the WPF model is 64.2% with an error rate of 35.8%, that is, assuming that the correct domain name is always contained in R. Applying the WPF model to our actual dataset results in an accuracy of 55.4%. For parameter h of Section IV-F, we have set this variable to 6, resulting into a total of 12 samples per website (2 per browser).

In the end, combining the IP address I and the traffic trace T will result into a correct and incorrect prediction with respectively 56.6%[13] and 9.6% of the websites, outperforming previous state-of-the-art work of [44]. It is however important to note that inferring a single domain from a single IP in [44] is based on a pre-processed dataset of the Alexa top 1M websites, while our experiment does not require such set and performs the method directly on the corresponding IP address. For comparison, guessing the domain name randomly from all SANs will result in a correct prediction for 19.4% of the websites. Finally, Figure 3 also shows a linear increase for all types listed, meaning that Open-Knock is not affected by the ranking (0 to 6000) of the domain name.

A valid consideration of our experiment is that we immediately start our identification process after receiving the IP address of the DNS response. However an ISP may only handle so much data at any given time, and may therefore extract the IP address and run the process after some time has

[11] Non-GDPR compliant websites often deny access to European consumers.

[12] I.e. the website is fingerprintable i.f.f the resulting list R is not empty, regardless of whether the correct domain name is contained in R.

[13] Includes the results of our WPF model.

passed. To demonstrate that our approach will still achieve acceptable results in this scenario, we re-run the experiment with the same IP addresses (without WPF model) but after a period of 14 days. Here, the number of correct predictions has only dropped slightly from 50.5% to 47.9%, while the number of incorrect predictions have remained nearly equal (9.2% to 9.1%). The number of IP addresses that return an empty SAN (from 29.3% to 31.8%) is the main reason of the decrease in correct predictions. We conclude that the time period of 14 days has only a minor impact on the results. Although comparing these results to previous work of [44] is difficult, as their dataset is generated within 2 hours and is not publicly available. It is therefore unclear whether the dataset in [44] holds up after 14 days, as IP rotations of DNS load balancers are common over time.

The dataset of our experiment and an online demo tool of Open-Knock are available for the public to use.

V. Limitations

We assumed that a zero-rated service consists of multiple unique domain names that are only used for that specific service, which is sometimes not the case. A realistic example for instance, are Facebook and Instagram which are 2 distinctive and unique services but the IP addressed used by the mobile app of Facebook are sometimes used by the desktop version of Instagram, and vice versa. This makes it difficult to choose whether a certain domain or IP should be zero-rated. Nevertheless, this does not entirely discredit our findings as the majority of the large commercial websites and mobile apps have their own unique domain name, that resolves to a set of unique IP addresses. Moreover, the limitation of deployed CDNs is also present in many current implementations of zero-rating by ISPs, and therefore still remains an unsolved problem today.

In the context of zero-rating, we also acknowledge the existence of tunneling in which consumers traffic is tunneled through a zero-rated server. For instance, the consumer may setup a webproxy that uses the zero-rated website as a platform to communicate with the destination non zero-rated website. Practical use-cases of such approach are always dependent on the zero-rated platform. E.g. on Twitter, tweets with raw TLS record data are sent by a custom client, while another webproxy extracts the data of these tweets and re-routes it to the appropriate non zero-rated website. Although performing this free-riding attack is network expensive, it nevertheless allows consumers to utilize a zero-rated low bandwidth channel. Besides Open-Knock, this limitation is also present in all previous studies discussed in Table I.

VI. Future Work

It is important to note that, although we only used SANs and reverse DNS as the input for our Open-Knock method, additional domain names may also be extracted from other sources. For instance, we could cache domain names from older look-ups or use optional pre-crawled domain names such as in [44]. Additionally, combinations with the DoH fingerprinting work of [52] may have a beneficial outcome on the accuracy of Open-Knock. Finally, a higher number of webpage samples (h) in Section IV-F or a longer delay q in Section IV-E may substantially increase the accuracy of our method, although future experiments would be useful to measure the exact impact.

VII. Conclusion

We comprehensively examined existing zero-rating approaches for ISPs and DPI vendors and showed that the majority of the approaches are unsustainable due to free-riding attacks and encrypted protocols such as ESNI (ECHO) and DoH. Based on industrial patents and academic studies, we rigorously discussed the concerns and flaws in current zero-rating implementations and presented solutions to solve these challenges. More specifically, we proposed Open-Knock, a method that is able to convert an IP address to a domain name such that it can be applied efficiently in zero-rating policies. Our experimental analysis demonstrates that Open-Knock outperforms previous state-of-the-art methods, when comparing performance and accuracy. Additionally, it attempts to protect the ISPs against free-riding attacks while leaving modern encryption protocols untouched and thus retain the privacy of the consumer. Although many of the previous limitations were thwarted, there are still open problems to be solved such as the use of tunneling and shared TLS certificates. We believe future studies should look into combining existing WPF techniques in order to achieve improved results for zero-rating implementations.

Acknowledgements

This research was funded in part by the Bijzonder Onderzoeksfonds (BOF) of Hasselt University. Finally, we thank Balazs Nemeth and Pieter Robyns for sharing their in-depth knowledge. We also thank the reviewers for their invaluable feedback.

References

[1] "The net neutrality rules that protect the open internet are in danger of being dismantled," accessed: 2019-07-10. [Online]. Available: https://www.google.com/takeaction/action/net-neutrality/

[2] "Unlimited video streaming with binge on^{TM}," accessed: 2019-07-10. [Online]. Available: https://www.t-mobile.com/offers/binge-on-streaming-video

[3] "Wikipedia zero," accessed: 2019-07-10. [Online]. Available: https://foundation.wikimedia.org/wiki/Wikipedia_Zero

[4] "Dns-over-https (doh)," June 2019, accessed: 2019-07-12. [Online]. Available: https://developers.google.com/speed/public-dns/docs/doh/

[5] "Percentage of web pages loaded by firefox using https," July 2019, accessed: 2019-07-16. [Online]. Available: https://letsencrypt.org/stats/

[6] R. Al-Kabra, R. SINHA, P. K. Bodiga, I. Ahamed, and J. Morrow, "Identifying user intention from encrypted browsing activity," 05 2019, US2019/0130036 (patent), T-Mobile USA, Inc.

[7] K. Al-Naami, A. El Ghamry, M. S. Islam, L. Khan, B. M. Thuraisingham, K. W. Hamlen, M. Alrahmawy, and M. Rashad, "Bimorphing: A bi-directional bursting defense against website fingerprinting attacks," *IEEE Transactions on Dependable and Secure Computing*, pp. 1–1, 2019.

[8] K. Al-Naami, S. Chandra, A. Mustafa, L. Khan, Z. Lin, K. Hamlen, and B. Thuraisingham, "Adaptive encrypted traffic fingerprinting with bi-directional dependence," in *Proceedings of the 32nd Annual Conference on Computer Security Applications*. ACM, 2016, pp. 177–188.

[9] H. F. Alan and J. Kaur, "Client diversity factor in https webpage fingerprinting," in *Proceedings of the Ninth ACM Conference on Data and Application Security and Privacy*, ser. CODASPY '19. ACM, 2019, pp. 279–290. [Online]. Available: http://doi.acm.org/10.1145/3292006.3300045

[10] D. Barr, “Common dns operational and configuration errors,” Internet Requests for Comments, RFC Editor, RFC 1912, February 1996, http://www.rfc-editor.org/rfc/rfc1912.txt. [Online]. Available: http://www.rfc-editor.org/rfc/rfc1912.txt

[11] I. N. Bermudez, M. Mellia, M. M. Munafo, R. Keralapura, and A. Nucci, “Dns to the rescue: Discerning content and services in a tangled web,” in *Proceedings of the 2012 Internet Measurement Conference*, ser. IMC ’12. ACM, 2012, pp. 413–426. [Online]. Available: http://doi.acm.org/10.1145/2398776.2398819

[12] K. Borgolte, T. Chattopadhyay, N. Feamster, M. Kshirsagar, J. Holland, A. Hounsel, and P. Schmitt, “How dns over https is reshaping privacy, performance, and policy in the internet ecosystem,” *Performance, and Policy in the Internet Ecosystem (July 27, 2019)*, 2019.

[13] Z. Chai, A. Ghafari, and A. Houmansadr, “On the importance of encrypted-sni (ESNI) to censorship circumvention,” in *9th USENIX Workshop on Free and Open Communications on the Internet (FOCI 19)*. Santa Clara, CA: USENIX Association, 2019. [Online]. Available: https://www.usenix.org/conference/foci19/presentation/chai

[14] H. Cheng and R. Avnur, “Traffic analysis of ssl encrypted web browsing,” 1998. [Online]. Available: https://pdfs.semanticscholar.org/1a98/7c4fe65fa347a863dece665955ee7e01791b.pdf

[15] G. Cherubin, J. Hayes, and M. Juarez, “Website fingerprinting defenses at the application layer,” *Proceedings on Privacy Enhancing Technologies*, vol. 2017, no. 2, pp. 186 – 203, 2017. [Online]. Available: https://content.sciendo.com/view/journals/popets/2017/2/article-p186.xml

[16] D. R. Choffnes, P. Gill, and A. Mislove, “An empirical evaluation of deployed dpi middleboxes and their implications for policymakers,” 2017. [Online]. Available: https://www.semanticscholar.org/paper/An-Empirical-Evaluation-of-Deployed-DPI-Middleboxes-Choffnes-Gill/904c6d870b994896515d5de7b292d1143e3f482b

[17] C. Cimpanu, “Kazakhstan government is now intercepting all https traffic,” July 2019, accessed: 2019-07-22. [Online]. Available: https://www.zdnet.com/article/kazakhstan-government-is-now-intercepting-all-https-traffic/

[18] ——, “Uk isp group names mozilla ’internet villain’ for supporting ’dns-over-https’,” July 2019, accessed: 2019-07-10. [Online]. Available: https://www.zdnet.com/article/uk-isp-group-names-mozilla-internet-villain-for-supporting-dns-over-https/

[19] D. Cooper, S. Santesson, S. Farrell, S. Boeyen, R. Housley, and W. Polk, “Internet x.509 public key infrastructure certificate and certificate revocation list (crl) profile,” Internet Requests for Comments, RFC Editor, RFC 5280, May 2008, http://www.rfc-editor.org/rfc/rfc5280.txt. [Online]. Available: http://www.rfc-editor.org/rfc/rfc5280.txt

[20] X. de Carné de Carnavalet and M. Mannan, “Killed by proxy: Analyzing client-end tls interception software,” in *Proceedings of the 24th Annual Network and Distributed System Security Symposium (NDSS 2016), Internet Society*, 01 2016. [Online]. Available: http://doi.acm.org/10.14722/ndss.2016.23374

[21] A. Delignat-Lavaud and K. Bhargavan, “Network-based origin confusion attacks against https virtual hosting,” in *Proceedings of the 24th International Conference on World Wide Web*, ser. WWW ’15. International World Wide Web Conferences Steering Committee, 2015, pp. 227–237. [Online]. Available: https://doi.org/10.1145/2736277.2741089

[22] K. V. der Stadt, “Onbeperkt dataverbruik van favoriete app soms toch aangerekend door foute metingen (dutch),” August 2019, accessed: 2019-08-31. [Online]. Available: https://datanews.knack.be/ict/nieuws/onbeperkt-dataverbruik-van-favoriete-app-soms-toch-aangerekend-door-foute-metingen/article-news-1503265.html

[23] M. Di Martino, P. Quax, and W. Lamotte, “Realistically fingerprinting social media webpages in https traffic,” in *Proceedings of the 14th International Conference on Availability, Reliability and Security*, ser. ARES 2019. ACM, 2019.

[24] D. Dixon, “Mozilla releases research results: Zero rating is not serving as an on-ramp to the internet,” July 2017, accessed: 2019-07-10. [Online]. Available: https://blog.mozilla.org/blog/2017/07/31/mozilla-releases-research-results-zero-rating-not-serving-ramp-internet/

[25] L. Dixon, T. Ristenpart, and T. Shrimpton, “Network traffic obfuscation and automated internet censorship,” *IEEE Security Privacy*, vol. 14, no. 6, pp. 43–53, Nov 2016.

[26] R. Dubin, A. Dvir, O. Pele, and O. Hadar, “I know what you saw last minute—encrypted http adaptive video streaming title classification,” *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 12, pp. 3039–3049, Dec 2017.

[27] I. Duvdevani and D. S. Naar, “Methods and systems for handling requests regarding zero-rating,” US2018/0048729A1 (patent), 08, 2016. [Online]. Available: https://worldwide.espacenet.com/publicationDetails/biblio?CC=US&NR=2018048729A1&KC=A1&FT=D#

[28] D. Eastlake, “Transport layer security (tls) extensions: Extension definitions,” Internet Requests for Comments, RFC Editor, RFC 6066, January 2011, http://www.rfc-editor.org/rfc/rfc6066.txt#section-3.

[29] J. L. García-Dorado, J. Ramos, M. Rodríguez, and J. Aracil, “Dns weighted footprints for web browsing analytics,” *Journal of Network and Computer Applications*, vol. 111, pp. 35 – 48, 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1084804518300894

[30] P. Hoffman and P. McManus, “Dns queries over https (doh),” Internet Requests for Comments, RFC Editor, RFC 8484, October 2018.

[31] R. Holz, L. Braun, N. Kammenhuber, and G. Carle, “The ssl landscape: A thorough analysis of the x.509 pki using active and passive measurements,” in *Proceedings of the 2011 ACM SIGCOMM Conference on Internet Measurement Conference*, ser. IMC ’11. New York, NY, USA: Association for Computing Machinery, 2011, p. 427–444. [Online]. Available: https://doi.org/10.1145/2068816.2068856

[32] M. Imani, M. S. Rahman, and M. Wright, “Adversarial traces for website fingerprinting defense,” in *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS ’18. ACM, 2018, pp. 2225–2227. [Online]. Available: http://doi.acm.org/10.1145/3243734.3278493

[33] M. Juarez, S. Afroz, G. Acar, C. Diaz, and R. Greenstadt, “A critical evaluation of website fingerprinting attacks,” in *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS ’14. ACM, 2014, pp. 263–274. [Online]. Available: http://doi.acm.org/10.1145/2660267.2660368

[34] A. M. Kakhki, F. Li, D. Choffnes, E. Katz-Bassett, and A. Mislove, “Bingeon under the microscope: Understanding t-mobiles zero-rating implementation,” in *Proceedings of the 2016 Workshop on QoE-based Analysis and Management of Data Communication Networks*, ser. Internet-QoE ’16. ACM, 2016, pp. 43–48. [Online]. Available: http://doi.acm.org/2940136.2940140

[35] M. Kirchler, D. Herrmann, J. Lindemann, and M. Kloft, “Tracked without a trace: Linking sessions of users by unsupervised learning of patterns in their dns traffic,” in *Proceedings of the 2016 ACM Workshop on Artificial Intelligence and Security*, ser. AISec ’16. New York, NY, USA: ACM, 2016, pp. 23–34. [Online]. Available: http://doi.acm.org/10.1145/2996758.2996770

[36] A. Klein and B. Pinkas, “Dns cache-based user tracking,” in *Proceedings of the 27th Annual Network and Distributed System Security Symposium (NDSS 2019), Internet Society*, 2019. [Online]. Available: https://dx.doi.org/10.14722/ndss.2019.2318

[37] V. Le Pochat, T. Van Goethem, S. Tajalizadehkhoob, M. Korczyński, and W. Joosen, “Tranco: A research-oriented top sites ranking hardened against manipulation,” in *Proceedings of the 26th Annual Network and Distributed System Security Symposium (NDSS 2018), Internet Society*, vol. abs/1806.01156, 2018, pp. 1–15. [Online]. Available: https://lirias.kuleuven.be/retrieve/529579Dndss19-final386.pdf

[38] F. Li, A. M. Kakhki, D. Choffnes, P. Gill, and A. Mislove, “Classifiers unclassified: An efficient approach to revealing ip traffic classification rules,” in *Proceedings of the 2016 Internet Measurement Conference*, ser. IMC ’16. ACM, 2016, pp. 239–245. [Online]. Available: http://doi.acm.org/10.1145/2987443.2987464

[39] F. Li, A. Razaghpanah, A. M. Kakhki, A. A. Niaki, D. Choffnes, P. Gill, and A. Mislove, “Liberate, (n): A library for exposing (traffic-classification) rules and avoiding them efficiently,” in *Proceedings of the 2017 Internet Measurement Conference*, ser. IMC ’17. ACM, 2017, pp. 128–141. [Online]. Available: http://doi.acm.org/10.1145/3131365.3131376

[40] M. Liberatore and B. N. Levine, “Inferring the source of encrypted http

connections," in *Proceedings of the 13th ACM Conference on Computer and Communications Security*, ser. CCS '06. ACM, 2006, pp. 255–263. [Online]. Available: http://doi.acm.org/10.1145/1180405.1180437

[41] B. Miller, L. Huang, A. D. Joseph, and J. D. Tygar, "I know why you went to the clinic: Risks and realization of https traffic analysis," in *Privacy Enhancing Technologies*, E. De Cristofaro and S. J. Murdoch, Eds. Cham: Springer International Publishing, 2014, pp. 143–163.

[42] G. Ollmann, "Dpi goes blind as encryption adoption increases," June 2016, accessed: 2019-07-10. [Online]. Available: https://www.vectra.ai/blogpost/dpi-goes-blind-as-encryption-adoption-increases

[43] A. Panchenko and F. Lanze, "Website fingerprinting at internet scale," in *Proceedings of the 24th Annual Network and Distributed System Security Symposium (NDSS 2016), Internet Society*, 2016. [Online]. Available: http://dx.doi.org/10.14722/ndss.2016.2347

[44] S. Patil and N. Borisov, "What can you learn from an ip?" in *Proceedings of the Applied Networking Research Workshop*, ser. ANRW '19. ACM, 2019, pp. 45–51. [Online]. Available: http://doi.acm.org/10.1145/3340301.3341133

[45] S. Perez, "Twitter's 'zero' service lets emerging markets tweet for free," 05 2014, accessed: 2019-07-09. [Online]. Available: https://techcrunch.com/2014/05/29/twitters-emerging-market-strategy-includes-its-own-version-of-a-facebook-zero-like-service-called-twitter-access/

[46] M. Prince, "Encrypting sni: Fixing one of the core internet bugs," September 2018, accessed: 2019-07-12. [Online]. Available: https://blog.cloudflare.com/esni/

[47] E. Rescorla, K. Oku, N. Sullivan, and C. Wood, "Encrypted server name indication for tls 1.3," Working Draft, Internet-Draft draft-ietf-tls-esni-04, July 2019, http://www.ietf.org/internet-drafts/draft-ietf-tls-esni-04.txt. [Online]. Available: http://www.ietf.org/internet-drafts/draft-ietf-tls-esni-04.txt

[48] ——, "Encrypted server name indication for tls 1.3," Working Draft, IETF Secretariat, Internet-Draft draft-ietf-tls-esni-06, March 2020, http://www.ietf.org/internet-drafts/draft-ietf-tls-esni-06.txt. [Online]. Available: http://www.ietf.org/internet-drafts/draft-ietf-tls-esni-06.txt

[49] V. Rimmer, D. Preuveneers, M. Juarez, T. Van Goethem, and W. Joosen, "Automated website fingerprinting through deep learning," in *NDSS*, 02 2018.

[50] G. Robertson, "Why the fall of net neutrality could smother innovation," December 2017, accessed: 2019-07-10. [Online]. Available: https://www.nginx.com/blog/fall-of-net-neutrality-will-smother-innovation/

[51] W. M. Shbair, T. Cholez, J. François, and I. Chrisment, "Improving sni-based https security monitoring," in *2016 IEEE 36th International Conference on Distributed Computing Systems Workshops (ICDCSW)*, June 2016, pp. 72–77.

[52] S. Siby, M. Juarez, C. Diaz, N. Vallina-Rodriguez, and C. Troncoso, "Encrypted dns $-->$ privacy? a traffic analysis perspective," 06 2019, (to appear).

[53] P. Sirinam, M. Imani, M. Juarez, and M. Wright, "Deep fingerprinting: Undermining website fingerprinting defenses with deep learning," in *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS '18. New York, NY, USA: ACM, 2018, pp. 1928–1943.

[54] Statoperator, "Https usage statistics on top 1m websites," July 2019, accessed: 2019-07-16. [Online]. Available: https://statoperator.com/research/https-usage-statistics-on-top-websites/

[55] N. Sullivan, "Dns-over-https will be rolled out by default (twitter: grittygrease)," September 2019, accessed: 2019-09-12. [Online]. Available: https://twitter.com/grittygrease/status/1170077782417666048

[56] C. V. Wright, S. E. Coull, and F. Monrose, "Traffic morphing: An efficient defense against statistical traffic analysis," in *Proceedings of the Network and Distributed System Security Symposium (NDSS 2009), Internet Society*, 02 2009.

[57] G. Venkatesh and L. Meixing, "Verification of server name in a proxy device for connection requests made using domain names," US2017/0374017A1 (patent), 06, 2016. [Online]. Available: https://worldwide.espacenet.com/publicationDetails/biblio?CC=US&NR=2017374017A1&KC=A1&FT=D

[58] N. Weaver, C. Kreibich, and V. Paxson, "Redirecting dns for ads and profit." *FOCI*, vol. 2, pp. 2–3, 2011.

[59] Y. Yiakoumis, S. Katti, and N. McKeown, "Neutral net neutrality," in *Proceedings of the 2016 ACM SIGCOMM Conference*, ser. SIGCOMM '16. New York, NY, USA: ACM, 2016, pp. 483–496. [Online]. Available: http://doi.acm.org/10.1145/2934872.2934896