IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received April 26, 2022, accepted May 11, 2022, date of publication May 16, 2022, date of current version May 26, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3175497

Summary of DNS Over HTTPS Abuse

KAREL HYNEK[1,2], DMITRII VEKSHIN[1,3], JAN LUXEMBURK[1,2], TOMAS CEJKA[1,2], AND ARMIN WASICEK[3], (Senior Member, IEEE)

[1]Faculty of Information Technology, Czech Technical University in Prague (CTU), 160 00 Prague, Czech Republic
[2]CESNET z.s.p.o., 160 00 Prague, Czech Republic
[3]Avast Software s.r.o., 140 00 Prague, Czech Republic

Corresponding author: Karel Hynek (hynekkar@cesnet.cz)

This work was supported in part by the Ministry of Interior of the Czech Republic (Flow-Based Encrypted Traffic Analysis) under Grant VJ02010024, in part by the Grant Agency of the Czech Technical University in Prague (CTU) through Ministry of Education Youth and Sports (MEYS) of the Czech Republic under Grant SGS20/210/OHK3/3T/18, in part by the European Union's Horizon 2020 Research and Innovation Program under Grant 833418, and in part by Avast Software s.r.o.

ABSTRACT The Internet Engineering Task Force adopted the DNS over HTTPS protocol in 2018 to remediate privacy issues regarding the plain text transmission of the DNS protocol. According to our observations and the analysis described in this paper, protecting DNS queries using HTTPS entails security threats. This paper surveys DoH related research works and analyzes malicious and unwanted activities that leverage DNS over HTTPS and can be currently observed in the wild. Additionally, we describe three real-world abuse scenarios observed in the web environment that reveal how service providers intentionally use DNS over HTTPS to violate policies. Last but not least, we identified several research challenges that we consider important for future security research.

INDEX TERMS Detection, DNS over HTTPS, hidden communication, IP flow, malware, network traffic, security threats.

I. INTRODUCTION

The DNS over HTTPS (DoH) [1] protocol has been recently developed to remediate the privacy issues of the Domain Name System (DNS) [2]. DNS transmits queries in plain text, and these queries can reveal sensitive information like a user's browsing habits. The main motivation for DoH is to limit the users' surveillance and protect them from possible profiling of their activities (e.g., for targeted advertising). Despite that the protocol specification has only been published in 2018, it has already spread vastly.

Transport Layer Security (TLS) protocol is the encryption protocol used in HTTPS communication and thereby became the de-facto standard for encrypted communication between clients and servers on the web. With TLS, a client and a server conceal the content of their communication to third parties. Before connecting to an HTTPS web server, a client typically needs to resolve the server's domain name, which has been done in plain text using DNS. The new step that DoH takes is to embed DNS queries in the HTTPS protocol. Thus, the entire transferred content, including queried domain names and answered IP addresses, is now concealed. Contrary to traditional DNS, where queries and translated domain names are visible to on-path entities, the content of the DoH traffic can be read by the client and its central DoH provider only. Third parties, including network operators and various network security tools like intrusion detection systems, cannot analyze the traffic without enforced interception of the communication using a decryption proxy server (which is not feasible in many environments).

The fact that DoH improves user privacy for legitimate traffic is undoubted. However, by building on the design principles of DNS, DoH also inherits some of its security issues. For instance, DNS has been exploited as a hidden communication channel (also known as the covert channel) in the past, for example, in DNSMessegner malware [3]. DoH amplifies this problem because its encryption prevents any analysis of the traffic content, which, in this case, would be the queried domain names and associated metadata. Since DoH support has already been added to many existing software applications [4], we expect attackers to leverage DoH to conceal their activities. The first step is an analysis of existing software and proof of concept codes to identify possible abuse techniques.

This paper was created by a collaboration of three organizations: Czech Technical University in Prague, CESNET (Czech national research and education network operator),

The associate editor coordinating the review of this manuscript and approving it for publication was Leandros Maglaras.

and Avast software, a global security company. It focuses on summarizing the current knowledge of DoH abuse, including three unpublished abuse scenarios. By monitoring DoH usage in the wild and manual inspection of DoH script samples and libraries, we have discovered leveraging of DoH for malicious purposes in web development – all utilize encryption to cover and retain service availability after their discovery and appearance on blocklists. The increased adoption of encrypted traffic is one of the biggest challenges that the network security field currently faces. DoH accelerates this trend for DNS traffic. The main contributions of the paper can be summarized as follows:

1) We provide survey of DoH related research.
2) We analyzed multiple public sources (such as Github, malware analysis research blogs, and VirusTotal) to list and taxonomize the DoH presence in current malware families and proof-of-concept code samples.
3) We described novel and previously unpublished abuse scenarios of DoH in the web environment. To the best of our knowledge, this is the first described real-world observation of DoH abuse in the web environment.
4) Based on the description of malicious use, we defined several research challenges that we consider essential for future security research.

The paper is organized as follows: Section II provides a brief introduction to the DoH protocol, Section III summarizes published research studies related to DoH, Section IV summarizes the knowledge about software abusing DoH and taxonomize it, Section V describes abuse scenario during our threat monitoring of the web environment, Section VI highlights research challenges that arise from DoH usage and its current or possible abuse, and final Section VII concludes the paper.

II. TECHNICAL DETAILS OF DoH PROTOCOL

In this section, we provide essential information about the DoH protocol specification and its packet-level behavior.

A. DoH SPECIFICATION

The IETF adopted the DoH protocol as an RFC document (RFC 8484 [1]) in 2018. Currently, there are two significantly different implementations. The RFC 8484 compliant approach uses classic DNS ''Wireformat'' [2] encapsulated in the HTTPS protocol. The messages are transferred either by HTTP GET or POST requests. The other approach uses DNS messages encoded in the JSON format described in RFC 8427 [5]. The JSON data are then transferred via HTTPS GET. Currently, most of the DNS providers (around 90%) support the ''Wireformat'', either HTTPS GET or POST version [6]. The JSON-based DoH is supported by around 30% of the DNS providers [6]. In practice, all of the DoH enabled browsers and most of the other performance-oriented DoH clients use RFC 8484 compliant Wireformat messages together with the HTTPS POST method.

The JSON approach also has its merits. The main reason to encode the DNS query in a JSON is to increase the readability and easy data manipulation based on text-based messages. According to our observation, JSON is used primarily for a single query by applications where performance and short response time are not a priority.

B. DoH FROM THE NETWORK MONITORING POINT OF VIEW

DoH follows the classic request-response scheme, with expected differences across HTTP protocol versions. Even though HTTP 1.1 is not officially recommended by RFC [1] due to performance reasons, most resolvers and browsers support it. The biggest performance bottleneck of HTTP 1.1 is the missing support of multiple concurrent requests within a single connection; therefore, it always has to wait for the response before sending the following query. According to our observations (in Chrome version 94,[1] and Firefox 91[2]), browsers reduce the performance penalty by creating multiple parallel connections (usually two). By switching between connections, they can perform concurrent requests. According to RFC 8484, each packet contains only one DNS query or response. Thus, network observers can reliably count the number of queries/responses transferred in the encrypted channel [7]. Apart from that, no other information can be directly obtained from the network packets due to the TLS encryption.

From the packet-level perspective, DoH looks similar to any other HTTPS communication. It establishes a connection on port 443, performs a TLS handshake, and transfers encrypted data. This allows DoH to effectively bypass DNS filters and other protections that analyze DNS queries' content. A typical system that relies on DNS analysis is a parental control application that prevents connections to certain websites such as social media or games by selectively blocking DNS requests [8]. However, as discussed in Section III-D1, proper DoH recognition is challenging task, which needs to employ sophisticated ML model. Currently, we are not aware of any commercial product that would use statistical methods or ML for DoH recognition and blocking.

III. RELATED RESEARCH ON DoH

DoH is a still relatively novel technology, which is waiting for mass adoption; however, there are already some published papers that target various aspects of it. In this section, we survey related work based on four perspectives: A) Performance Perspective, B) Adoption Perspective, C) Privacy Perspective, and D) Security Perspective. For the research of related work, we used the following indexing engines: ACM Digital Library,[3] IEEE Explore,[4] Scopus,[5] and Google Scholar.[6]

[1] https://chromereleases.googleblog.com/2021/09/stable-channel-update-for-desktop_30.html
[2] https://www.mozilla.org/en-US/firefox/91.0/releasenotes/
[3] https://dl.acm.org
[4] https://ieeexplore.ieee.org
[5] https://scopus.com
[6] https://scholar.google.com

A. PERFORMANCE PERSPECTIVE

The latency of DNS protocol directly impacts the performance of networking applications [9]. Therefore, many researchers measured the performance consequences of DoH deployment. These studies are summarized in Table 1.

One of the first DoH latency measurements was published by McManus [10] from Mozilla in 2018, showing that the average additional latency caused by DoH is only 6 ms. The following study created by Böttger *et al.* [11] focused on DoH overhead compared to traditional DNS. Their results show that DoH adds significant latency when the connection is used for a single query. However, when DoH connection is reused for multiple queries, the additional latency is negligible. Another study performed by Hounsel *et al.* [12] shows that DoH latency and reliability strongly depend on the selected resolver. This is also supported by Jerabek *et al.* [13] who studied DoH resolver behavior and the distribution of DoH packet sizes depending on used resolvers. According to their results, some DoH resolvers use long HTTP headers resulting in larger packets and thus bigger overhead.

A more extensive study was performed by Chhabra *et al.* [14], who studied DoH performance impact across the world. Their results show that users from higher-income countries with higher quality internet infrastructures are less likely to experience slower performance caused by DoH, resulting in a disproportionate impact on users from countries with lower economic capacity. Their findings are also supported by the studies performed by Hounsel *et al.* [15], Borgolte *et al.* [8] and Mbewe and Chavula [16], who also show that DoH has a negligible impact in good network condition. According to these studies [8], [15], [16], traditional DNS significantly outperforms DoH when dealing with congested or 3G mobile networks.

TABLE 1. Comparison of DoH performance related research. Measurement Setup — measurement data and its origin, Results — The main conclusions of the measurement about the DoH performance impact compared to traditional DNS.

Author	Year	Measurement Setup	Results
McManus [10]	2018	Firefox users	Negligible impact, added latency of 6 ms
Böttger et al. [11]	2019	Single client	Negligible impact on latency when reusing connection
Borgolte et al. [8]	2019	Self-emulated network conditions	Selective impact, depending on network conditions
Hounsel et al. [15]	2020	Self-emulated network conditions	Selective impact, depending on network conditions
Hounsel et al. [12]	2021	Generated via endpoints across North America	Selective impact, depending on used DoH resolver
Chhabra et al. [14]	2021	Worldwide measurement across 224 countries	Selective impact, depending on network conditions
Mbewe et al. [16]	2021	Generated via endpoints across Africa	Selective impact, depending on network conditions
Jerabek et al. [13]	2022	Generated, single location	Selective impact depending on used DoH resolver.

B. ADOPTION PERSPECTIVE

At the time of writing, DoH is supported (and sometimes enabled by default[7]) by most modern web browsers such as Chrome (since version 83[8]), Edge, Firefox, Opera, and Brave; a comprehensive evaluation of DoH support in web browsers can be found at zdnet.com [4]. There are also native resolvers with DoH support in Microsoft Windows [17] and modern GNU/Linux distributions (e.g., via systemd-resolved). DoH is supported by major domain name server software such as BIND (since version 9.17.10), KNOT resolver (since version 5.2.0), and Unbound (since version 1.12.0). There is a DoH proxy by Cloudflare called `cloudflared`. There are at least eight DoH client implementations and at least six server implementations known and listed at dnscrypt.info.[9]

The support of DoH by open resolvers was studied in 2019 by Deccio and Davis [18]. Their results show that the DoH adoption was very poor. From around 1.2 million open resolvers, only nine supported DoH. A later study in 2021 was carried out by Garcia *et al.* [6]. In this study, the authors scanned the entire IPv4 address range and found 931 addresses that successfully resolved DNS over HTTPS. Unfortunately, both studies measured adoption on different data (open resolvers vs. entire IPv4 address range); thus, we cannot conclude any adoption increase among service providers between 2019 and 2021.

The DoH adoption by users was also studied by Garcia *et al.* [6]. The paper presents three large datasets from a large European university, a large European internet service provider, and a global security company. The results show that the volume of DoH traffic increased during 2020; however, DoH remains relatively rare compared to traditional DNS. The summary of DoH adoption-related studies is shown in Table 2.

C. PRIVACY PERSPECTIVE

Since the primary benefit of DoH is the increased privacy of end-users [1], it has been thoroughly studied by many researchers. The privacy-focused studies are summarized in Table 3. Overall, there is a general scepticism [19], [20] about the sufficiency of DNS encryption for preserving users' privacy. Therefore, the DNS protocol privacy enhancement feature called EDNS padding [21] was introduced. Clients with DoH support send requests padded with random content to equalize the sizes of all packets. The padding reduces the possibility of side-channel information leakage.

Website fingerprinting is one of the possible attacks, which leverage the side-channel information. The fingerprinting attacks are built on the assumption that connection to each

[7]https://support.mozilla.org/en-US/kb/firefox-dns-over-https

[8]https://www.zdnet.com/article/chrome-83-released-with-enhanced-privacy-controls-tab-groups-feature/

[9]https://dnscrypt.info/implementations/

TABLE 2. Comparison of DoH adoption related research. Measurement Setup — measurement data and its origin, Results — The main conclusions about the DoH adoption measurement.

Author	Year	Measurement Setup	Results
Deccio et al. [18]	2019	Across Open Resolvers	Adoption < 1 ‰
Garcia et al. [6]	2021	IPv4 address space Traffic from 3 organizations	931 DoH capable IP addresses, Volume of DoH traffic is increasing, DoH is relatively rare

website generates a unique sequence packets' sizes, which the adversary can leverage to infer the transferred and encrypted content [22]. Bushart and Rossow [23] and Siby *et al.* [22] performed a website fingerprinting attack using DoH traffic only by leveraging the lack of EDNS padding. Authors point out that their approach requires fewer data to process while maintaining similar accuracy compared to traditional fingerprinting. Both papers also evaluated the traffic with the EDNS padding feature enabled, and they were still successful with more than 70% accuracy.

Hynek and Cejka [7] performed an experiment similar to the website fingerprinting; however, they aimed to infer actual queries inside a single DNS packet. They studied the shape of DoH traffic and showed that it is possible to identify the number of queries or used versions of the HTTP protocol. Moreover, they leveraged DoH packet sizes to infer queried domain names with accuracy 90% when using HTTP 1.1. However, their method proved unusable when the EDNS padding feature was enabled.

The downgrade privacy attack was studied by Huang *et al.* [24]. They performed a downgrade attack by blocking the DoH connection, forcing the browsers to roll back to traditional unencrypted DNS without any noticeable alert in the user interface. According to the study [24], browser vendors do not consider this attack as a vulnerability but rather a well-documented feature also described in RFC 8310 [25]. The impact of a downgrade attack could be reduced by proper notification about lost privacy; however, none of the browser vendors plan to integrate it [24].

Other privacy concern related to DoH is the data centralization by DoH providers and the possible correlation and misuse of clients' IP addresses and DNS requests. The data centralization is addressed by Oblivious DoH (ODoH) proposal [26], which uses an intermediate proxy for queries. The proxy would know the clients' IP addresses but cannot inspect the payload of packets. The resolvers can read the payload; however, the clients' IP addresses are hidden behind the proxy. Currently, the ODoH is in the state of RFC draft [27] with available proof-of-concept codes.[10]

D. SECURITY PERSPECTIVE

Studies [8], [28], and [29] on the impact of DoH mass deployment conclude that DoH is a security problem since

[10]https://github.com/cloudflare/odoh-go

TABLE 3. Comparsion of DoH privacy related research. The study scope abbreviation stands for: C — Correlation of encrypted and unencrypted DNS on recursive resolver, FP — Fingerprinting attack, DG — Downgrade attack, P — Proposal of novel technology.

Author	Year	Scope	Outcomes
Shulman et al. [20]	2014	C	Execution of correlation attack for domain inference.
Bushart et al. [23]	2019	FP	ML model for website recognition. 86.1% accuracy when no defence mechanism used.
Siby et al. [22]	2019	FP	ML model for website fingerprinting. 0.908 F1 score when no defence mechanism used
Hynek et al. [7]	2019	FP	ML model for queried domain name inference. 90.14% accuracy when no defence mechanism used
Huang et al. [24]	2020	DG	Execution of DoH downgrade attack in web browsers.
Singanamalla et al. [26]	2020	P	Proposal of Oblivious DoH to increase DoH users' privacy.

many existing automated network security tools rely on unencrypted DNS. Attackers can leverage the increased privacy of encrypted DNS to hide their malicious activities. Even though DoH provides confidentiality of resolution, it does not protect against subversion of DNS resolution (such as DNS cache poisoning) [30] and allows the creation of DNS tunnels [31]. The DoH studies from a security perspective can be divided into two categories: 1) Detection of DoH presence in the network and 2) Detection of malicous DoH. Studies from both categories are then summarized in Table 4.

1) DETECTION OF DoH PRESENCE IN NETWORK

DoH decreases visibility by automated network security tools [8]; therefore, detection of DoH presence can be considered viable for maintaining situational awareness of network operators and analysts. DoH traffic in the highly restricted network might indicate an attempt of policy violations or the presence of some unwanted software. Since DoH does not use any dedicated port number, it blends into other encrypted HTTPS traffic, making its recognition difficult. Majority of DoH can be blocked by filtering 443/TCP connections to well-known DoH providers (such as Google or Cloudflare). However, it is always possible to choose less known DoH resolver that anyone can deploy — there are already available open-source software capable of DoH to DNS translation. Moreover, according to Garcia *et al.* [6], there are hundreds of ''unknown'' DoH resolvers that do not appear on public DoH blocklists.

One of the first studies that proposed DoH detection by its traffic characteristics was published in 2020 by Vekshin *et al.* [32]. In this work, the authors trained several machine learning models to distinguish DoH connections from other traffic, achieving high accuracy of 99% (0.99 F1 score). According to Vekshin *et al.* [32], the most important traffic feature for the detection of DoH is the duration of

TABLE 4. Comparison of research considering DoH security. The abbrevations in scope column stand for: D — DoH detection, E — DoH exfiltration, S — Summary, R — Subversion of DNS resolution.

Author	Year	Scope	Method	Dataset	Outcomes
Vekshin et al. [32]	2020	D	AdaBoost	Custom	DoH detector with accuracy of 99% (0.99 F1 score) DoH client (browser) identification with accuracy of 99%.
Bumanglag [29]	2020	S	—	—	General discussion about the impact of DNS over HTTPS on cyber security. Survey of traditional DNS threats and their detection possibility in DoH.
MontazeriShatoori [31]	2020	D & E	Random Forest	DoHBrw-2020	DoH detector with 0.99 F1 score. DoH exfiltration detector with 0.99 F1 score.
Banadaki et al. [33]	2020	D & E	XGBoost	DoHBrw-2020	DoH detector with claimed 100% accuracy. DoH exfiltration detector with claimed 100% accuracy.
Singh et al. [34]	2020	E	Gradient Boosting	DoHBrw-2020	DoH exfiltration detector with claimed 100% accuracy.
Wu et al. [35]	2021	D	Autoencoder	Custom	DoH exfiltration detector with 98% accuracy.
Badhwar [30]	2021	R	—	—	Discussion about the subversion of DoH resolution and necessity to deploy it alongside with DNS security extension.
Casanova et al. [36]	2021	D	Bi-LSTM	DoHBrw-2020	DoH exfiltration detector with 99% accuracy.
Csikor et al. [37]	2021	D	Random Forest	Custom	DoH exfiltration detector with 0.97 F1 score, when no defence mechanisms used. DoH detection defence techniques, which drops the detector performace to unusable level.
Kwan et al. [38]	2021	E	Simple statistical	Custom	They achieved 94% accuracy by observing outgoing throughput.
Ding et al. [39]	2021	E	Autoencoder	DoHBrw-2020	DoH exfiltration detector with 0.99 F1 score.
Behnke et al. [40]	2021	D & E	LightGBM	DoHBrw-2020	DoH detector with 99% accuracy DoH exfiltration detector with claimed 100% accuracy.
Alenzi et al. [41]	2021	E	XGBoost	DoHBrw-2020	DoH exfiltration detector with 99% accuracy.
Zebin et al. [42]	2022	D & E	Balanced Stacked Random Forest	DoHBrw-2020	Single detector capable of distinquishing DoH exfiltration, DoH and non-DoH traffic with accuracy of 99%.
Zhan et al. [43]	2022	E	Random Forest	Custom	DoH exfiltration detector with 0.99 F1 score.

the connection, its burstiness, and the number of transferred packets. However, they worked only with browser-based DoH connections, leaving a single query DoH undetected. Following studies [31], [33], [35], [36], [40] also achieved similar results, proving that browser-based DoH has distinctive properties that can be leveraged for detection. Csikor *et al.* [37] expressed a concern about the DoH detection possibility, arguing that it can be misused for censorship by downgrade attack. Therefore, they have evaluated multiple DoH padding techniques, which modified the DoH traffic characteristics, making them similar to regular HTTPS. One of the evaluated techniques successfully degraded the performance of machine learning detectors to the level where its deployment would be impractical.

2) DETECTION OF MALICIOUS DoH

The traditional DNS abuse detection is a well-studied topic, which is targeted by many research works [44]–[51]. However, none of the mentioned work can be directly applied to DoH due to the added encryption.

The security-related research in the DoH area focuses mainly on data exfiltration. MontazeriShatoori *et al.* [31] analyzed the DoH tunneling approaches and the possibility of their detection. They created a dataset called DoHBrw-2020[11] and proved the usability of time-related features to detect DoH tunnels and reported an accuracy of almost 100% (F1 score 0.999).

Many studies [33], [34], [39]–[42] then used the DoHBrw-2020 to prove the possibility of malicious DoH detection with various machine learning approaches, all of them achieving very high accuracy above 99%. However, the DOHBrw-2020 consists of only lab-created traffic from tunneling tools that use traditional unencrypted DNS, translated into DoH using a proxy. The dataset does not include traffic from already DoH capable malware samples or exfiltration tools. These weaknesses were addressed by studies performed by Kwan *et al.* [38] and Zhan *et al.* [43]. Both studies focused on a more realistic scenario of DoH tunnel detection using a DoH capable exfiltration tool. Kwan *et al.* focused on simple detection techniques using only a single feature, such as throughput, and achieved 93% accuracy by observing only outgoing throughput. Zhan *et al.* [43] performed DoH based exfiltration between various locations worldwide. They tested multiple machine learning classifiers and achieved detection accuracy above 99%.

We are not aware of any study that focuses on detecting other malicious DoH than exfiltration, such as DoH Command and Control (C2) detection, which is described in Section IV-A. Compared to traditional DNS, the research targeting DoH abuse detection is still nascent mainly using lab-created DOHBrw-2020 dataset. Therefore in the following sections, we summarize the state of DoH abuse and point out the main research challenges that should be targeted by the research community.

IV. TAXONOMY OF DoH ABUSE: TOOLS & MALWARE

Since DoH is built upon the traditional DNS, the abuse possibilities of DoH can be derived from DNS protocol. According to the 2016 Cisco annual security report [52], 91.3% of malware families use DNS, and the number does not seem to be decreasing. DNS is primarily abused for accessing C2 infrastructure as well as data exfiltration. Incorporating DNS into malware's infrastructure increases its resilience against threat protection systems, for instance, when combined

[11] https://www.unb.ca/cic/datasets/dohbrw-2020.html

TABLE 5. Number of DoH-capable code samples/malware strains for each category.

Category	#	References
C2 Access and Communication	10	[55], [57]–[63], Novel abuse scenario described Sec. V-B,V-C
Covert Multipurpose Channels	4	[64]–[66], Novel abuse scenario described in Sec. V-A
Unaware Usage	Any SW with DNS	

with Domain Generation Algorithm (DGA) [53] and Fast Flux [54] techniques. The resilience of malware even increases when deploying these techniques via DoH due to added encryption.

Malware creators are aware of the advantage of encryption and have started to use it in order to avoid detection [55]. However, not every traditional DNS abuse technique can be applied to DoH. For example, DNS amplification, a common DDoS attack vector, is a widespread problem firstly described by Randal Vaughn [56]. Fortunately, DNS amplification cannot be performed with DoH. DNS amplification attacks spoof source IP addresses such that the DNS resolver's response is sent to the victim's system. DoH requires establishing a TCP connection; thus, source IP address spoofing is not possible as it is with DNS over UDP.

We have analyzed multiple public sources of information and related works (such as Github, malware analysis research blogs, and VirusTotal[12]) to summarize the state of DoH abuse. We divide known DoH abuse into three categories: 1) C2 Access and Communication, 2) Covert Channels, and 3) Unaware Usage. Table 5 summarizes the number of DoH abusing code/malware samples we are aware of for each category. The categories and the code/malware samples are described in further detail in the following sections.

A. C2 ACCESS AND COMMUNICATION

C2 communication is one of the most common abuses of unencrypted DNS. In the encrypted case, most malware use DoH only to gain access to the C2 infrastructure. C2 communication itself then continues via other protocols. An example of such usage is the `PsiXbot` malware. The analysis created by the Proofpoint threat insight team [57] reveals that `PsiXbot` uses the hardcoded `dns.google.com` resolver and issues a JSON-based DoH request via HTTP 1.1 to resolve a hardcoded C2 domain. After receiving the C2 server IP, the communication between C2 and malware uses HTTP, which is unencrypted. Interestingly, the HTTP payload is encrypted using the RC4 algorithm. Similarly, banking malware `FluBot`, which targets Android devices, also relies on DoH to access its C2 infrastructure [58]. Translating a domain name via DoH is not by itself abuse. However, the intent of hiding such communication in encryption to bypass detection systems is undoubtedly abuse.

Another case of DoH abuse was published by Huntress-labs [59] describing the JSON-based TXT request for DKIM using DoH via the `dns.google.com` domain resolver. The TXT answer contained the IP addresses of external servers for downloading another payload to complete the C2 access. Both approaches exploit the fact that Google DNS is the most popular DNS resolver [67];thus it is probably accessible.

Overall, we are currently aware of five approaches that gain access to the C2 infrastructure using DoH [55], [57]–[60], and all of them are slight modifications of the two mechanisms described above. All five approaches use the JSON API of DoH and they mostly use Google's DNS resolver. The only exception is the `Godlua` malware [55], which uses Cloudflare's DNS resolver.

Malware can also utilize DoH as a channel for the transmission of C2 commands. The `LSD` malware [61] uses DoH for accessing C2 infrastructure and downloading (via TXT records) a bootstrap script to connect to a crypto-mining pool proxy.

There are also other proof-of-concept (PoC) source codes that are — to our best knowledge — not yet deployed in any actual malware. One noteworthy PoC code is `godoh` [62], [68], which uses DoH via its JSON API to tunnel C2 conversations. A similar concept called `DoHC2` [63] was implemented for the adversary simulation and red team operations software Cobalt Strike.[13]

B. COVERT MULTIPURPOSE CHANNELS

Some solutions for covert channels natively support DoH. The `dnstt` [69] is tool capable of exfiltration via DoH. Similarly, the `DNSExfiltrator` [64] can upload files to the server via DoH with Google's or Cloudflare's resolvers. Ciampanu [70] reports that `DNSExfiltrator` is already used by the OilRig group, which is tracked as Advanced Persistent Threat group 34 (APT34). In addition, DoH tunnels are already covered in red team seminars and conferences like 44CON [65] or BruCON [66], where an Excel sheet downloads malware via a DoH tunnel.

Moreover, there are multiple solutions available on regular, unencrypted DNS, such as `Iodine` [71], `DNSCat` [72], or `TUNS` [73]. Even though these well-known and easy-to-use programs do not support DoH, they can extend their capabilities by running a DoH proxy.

Even though the DoH tunnel performance is reported to be slower than tunneling via traditional DNS [74], the tunnels can be established and are reported to work. Strikingly, DoH resolvers do not deploy any protection against DNS abuse because an unstealthy and evident DNS tunnel could be established via major DoH resolvers, like Google, Cloudflare, and AdGuard [31].

C. UNAWARE USAGE

For comprehensiveness, there is also a separate category, "Unaware Usage", which we have identified during the sur-

[12]www.virustotal.com

[13]https://www.cobaltstrike.com

TABLE 6. Used IP addresses for recognition of DoH connection during our finding of DoH-capable malicious software samples.

Resolver	IPv4	IPv6
Cloudflare	1.1.1.1,1.0.0.1, 104.16.248.248, 104.16.248.249, 104.16.249.248, 104.16.249.249	2606:4700:4700::1111, 2606:4700:4700::1001
Google	8.8.8.8,8.8.4.4	2001:4860:4860::8888, 2001:4860:4860::8844

vey and analysis. With the large-scale deployment of DoH in popular browsers and Operating Systems, malware DNS communication might get encrypted without the malware's intention or awareness of the encryption. Canonical examples are web browser extensions that call a browser API for domain resolution, or malware might use DoH because DoH is set as a default DNS method in the OS. As an example of the consequence, malware that can be easily detected at the network level by some typical DNS queries becomes automatically harder to detect due to encrypted communication, even though the malware itself is not aware of DoH.

From the network security perspective, these scenarios are the most challenging. We are not aware of any study that analyzed the detection possibility of malicious DNS traffic mixed with benign inside the same DoH connection. Untangling the mix is a challenging problem.

V. NOVEL OBSERVATIONS OF DoH ABUSE

As a part of a large global security company protecting hundreds of millions of endpoints, our laboratory has access to a continuous feed of suspicious software, malware, and malicious websites samples analyzed in a sandbox environment. The automated analysis pipeline allows the selection of particular malware samples for further inspection. We filtered malicious samples performing DoH based on port (443/TCP) and IP addresses of known Google and Cloudflare DoH resolvers which are written in Table 6. We decompiled or deobfuscated the source codes of found DoH-capable samples and manually analyzed them, looking for functions processing DoH requests and responses. In some web-based samples, we spotted unconventional and unpublished use of DoH by service providers to avoid DNS-based service blocking.

Many countries perform website censorship and blocking according to local laws. It is a common practice because our modern society considers many types of content as harmful and unacceptable. The prevention of access to some internet resources helps to fight against child pornography, copyright infringement, and many more. There are multiple ways of implementing the web content blocking [75], [76]. However, many countries implement it using DNS Tampering, i.e., a spoofed DNS answer can deny the existence of the domain name or redirect users to some block page (that can be operated by the government) with the reason of the website closure [76]. The DNS tampering procedure is depicted in Figure 1. Naturally, DoH effectively bypasses this blocking

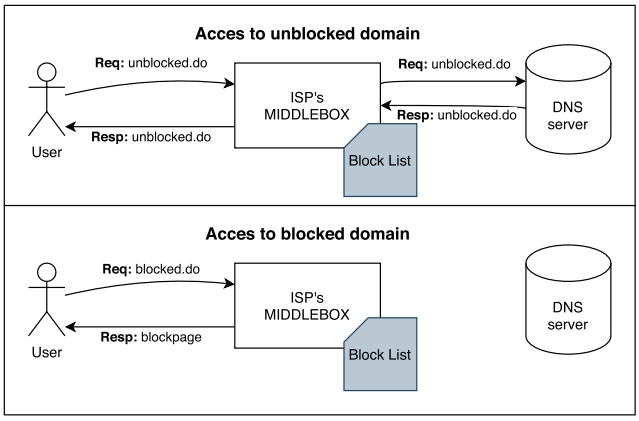

FIGURE 1. Scheme of DNS tampering procedure.

mechanism, which can be leveraged by service providers leaving users unaware of their illegal activity.

The rise of DoH support enabled malware authors to access an easy-to-use JSON-based DNS API through the browsers' JavaScript interpreter that can be leveraged in browser-based exploits. Specifically, multiple service providers (C2 services) were observed to take advantage of encryption and easy-to-use DNS-based C2 communication channels. All of them abuse DoH to avoid website censorship and blocking.

We are unaware of any previous study describing DoH abuse by service providers on the web, which is also a critical field related to computer security and network monitoring. Even though some of the identified threats are known or similar to traditional DNS threats, they appeared recently in the DoH domain. In this section, we present a real-world observation of their transfer into the encrypted domain, which proves an adoption of DoH abuse in web-based threats. The observations are organized in three abuse scenarios: 1) Client Modification to Access Blocked Websites, 2) DoH in Website Redirections, and 3) DoH Requests in Advertisements and Spam Campaigns.

A. ABUSE SCENARIO 1: CLIENT MODIFICATION TO ACCESS BLOCKED WEBSITES

The abuse scenario assumes two entities — client and server. The client wants to communicate with the server; however, direct communication is not allowed, and its prevention is implemented by DNS tampering on the local DNS recursor. The client is modified to use DoH to bypass blocking mechanisms and obtain the working server's IP address that allows direct communication.

Even though there is almost a universal support of DoH in web browsers, other types of programs still lack the support. The most straightforward modification is installing a DoH proxy that translates all local DNS requests into DoH. However, it requires much effort from the users, and we have already observed more user-friendly client modifications that use DoH only for accessing the blocked websites.

```
#b64decode domain
API = b64decode('aHR0...90LnR2')
def getIP (domain):
    #b64 encoded domain:
    #https://dns.google.com/resolve/type=A&name=
    URL='aHR0...25hbWU9'
    req = http.get(b64decode(URL)+domain);
    return json.parse(req)[IP]

def create_url(sid, season, episode):
    ipaddr = getIP(API);
    final_url = "http://"+ipaddr+"/"+sid+"/"\
    +season+"/"+episode
    return final_url
```

Listing 1. DoH usage example in Sdarot Kodi plugin.

A real-world example is `sdarot.tv`, an Israeli based website that provides video content. Due to the copyright violation, it was blocked by the Israeli government, and all local internet service providers have to prevent access by DNS Tampering [77]. However, the website is still flourishing due to the multiple non-browser clients and their modifications. Sdarot provides a plugin written in python for the home theater software Kodi, and its short and simplified code snipped can be found below in Listing 1. The plugin uses *base64* encoded domain names in the translation process. After the translation, all URLs contain IP addresses directly to avoid DNS resolvers of the operating system and ISP.

Sdarot also provides Android and Android TV applications that do not use DoH. However, the applications bypass the system settings and use the Google DNS servers instead of the local DNS recursor. In addition, we analyzed the decompiled Java code, and it indeed contained code for DoH JSON-based queries. Thus, the DoH support might be enrolled soon because the simple use of some foreign DNS resolvers is already insufficient in some states [78].

The Abuse Scenario 1 falls into a *Covert Multipurpose Channels* category of DoH abuse described in Section IV-B.

B. ABUSE SCENARIO 2: DoH IN WEBSITE REDIRECTIONS

The abuse scenario assumes three entities – client, server, and C2 domain. The client is redirected to the server or performs willing access. On the first visit, the server modifies the client's browser by installing a redirection mechanism. Later, the server is identified as malicious, and the DNS tampering technique prevents its access. Due to the installed modification, the browser recognizes the prevention access mechanism and performs a DoH request to the C2 domain. The response contains a functional landing domain of the server that allows its access.

During the monitoring of DoH usage in our laboratory, we found DoH requests created in web-based JavaScript by multiple websites. The websites use DoH for redirection to illegal online casinos targeting Russian citizens.

Since 2009, the gambling business has been banned in the Russian Federation, with a few gambling zones exceptions. As a result, all online casinos (even non-Russian) are

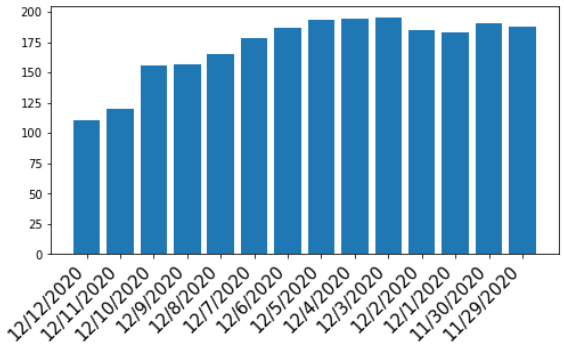

FIGURE 2. Number of unique domains pointing to single IP address of selected website with DoH redirection script according to Security Trails Passive DNS data.

prohibited in Russia. Even advertisement to gambling websites is considered illegal. The online gambling organizers risk a fine of up to 14,000 USD and website closure by the government. Despite the severe penalties, Russia's illegal gambling market is worth about 7.9 billion USD per year [79].

The online casinos are fighting the gambling ban by changing IP addresses and registering multiple domains. We have used the Security Trails Passive DNS system[14] to monitor a domain name of selected online casino IP address. As it can be seen in Figure 2, more then 100 domain names point to the same website according to the Passive DNS data.

The rapid domain name changing strategy is almost identical to malware C2 infrastructure, which uses DGA. However, the casinos depend on users, who are unwilling to test the connection to hundreds of domains. Therefore, there is a redirection infrastructure in place that ensures landing on the functional unblocked casino website.

In all observed JavaScript codes samples, which performed DoH requests, the redirection occurs in the web browsers as a JavaScript Service Worker — an API that allows websites to install JavaScript code into the browser. It is like a browser plugin that can run only on domains (and all its subdomains) that installed it. When the user accesses the page, the service worker is initiated and runs in the background, separate from other websites' JavaScript code. Even though the service worker API is limited, it can register callbacks for events such as "website fetch" and modify the content similarly as a proxy.[15]

The redirector service worker is installed in the browser when the user enters the casino website. Next time, when the user wants to access, the redirector activates. In all analyzed websites, the redirector issued a DoH TXT request to a C2 domain and got a *base64* encoded JSON object. The format of the TXT answer is shown in Listing 2. The array contains a redirection enable flag, body substring, and the functional landing domain. The body substring distinguishes between a government block page and the actual casino webpage. It is

[14]A system, which records the history of resolved domains and their belonging IP addresses. URL: https://securitytrails.com

[15]https://developer.mozilla.org/en-US/docs/Web/API/Service_Worker_API

```
{
    "1":1,
    "2":"VDuXmwmNZ",
    "3":"https://somedomain.com"
}
```

Listing 2. The example of the decoded TXT answer. 1 – is the enable flag, 2 – identifier for distinguishing between block pages and the correct output, 3 – redirection domain.

```
//response arg. contains the server response
//for previous GET query
function onWebsiteFetch(response){
   [enabled, check_string, domain] = get_domain();
   if(enabled and !response.contains(check_string))
      redirect(domain);
   else
      return response;
}
function get_domain() {
   //b64encoded DoH query to C2 domain
   resp = fetch(b64decode("aHR...bWU="))
   return json.parse(b64decode(resp)).txtContent
}
```

Listing 3. DoH redirector in Service Worker.

usually a short identifier that occurs in the body tag of the webpage.

The service workers scripts in four analyzed websites were very similar, with minor differences in function names, or used API, showing that all of them were implemented separately. The example of the observed redirection script is shown in Listing 3. According to the instruction from the C2, the service worker checks whether the domain is blocked. If not, the user proceeds to the webpage. In the other case, the user is redirected via JavaScript to the landing domain, and a new JavaScript Service Worker is installed. By this mechanism, users can remember only one URL (the first one they have visited) and are always redirected to the functional unblocked website. The whole redirection scheme is depicted in Figure 3.

We have analyzed selected C2 responses in time with the Security Trails Passive DNS System.[16] All unique domain names, that appeared in the responses between September 29th and November 20th in 2020 are shown in Figure 4. Overall, in the observed period, the landing domain name changed 35 times. It can be noticed that some of the landing domain names are very similar and differs in only a single character, which is sufficient for bypassing the DNS tampering.

We have found eight different C2 domains that redirect to more than 80 websites during our research. All of them targeted the Russian market and were related to gambling. However, the presented approach has enormous potential in more fields other than gambling. Unfortunately, the same scheme can be used in more severe cases like malware or even child pornography distribution. Besides, the presented DoH based redirection can potentially substitute the domain fronting [80] (a technique for censorship bypass utilizing infrastructure with multiple services), which is already banned by large CDN providers [81].

[16]https://securitytrails.com

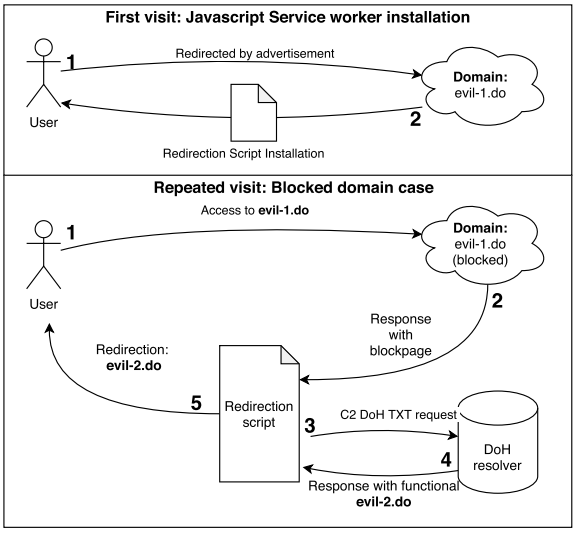

FIGURE 3. Redirection scheme.

The Abuse Scenario 2 falls into a *C2 Access and Communication* category of DoH abuse described in Section IV-A and its mass deployment can enable a hidden web (like the dark web). Websites could change their domains and IP addresses more rapidly (in a matter of minutes) without reduced comfort for users. The only problem is the first visit, which can be performed via advertising (as described) or other services that would query the C2 domain and provide the first redirect. The state authorities are almost defenseless against this redirection principle. The C2 domain might seem like a candidate for a weak spot because its inaccessibility would cause the collapse of the whole redirection infrastructure. However, when the C2 domain is accessed solely by DoH, the access can be prevented only by the DoH resolver or by the TLD[17] operator. Even though the DoH provider can technically prevent access to a particular domain, users can always use a different one that does not perform blocking. TLD operators can perform forced domain shut-down. However, it is usually complicated to achieve.

C. ABUSE SCENARIO 3: DoH REQUESTS IN ADVERTISEMENTS AND SPAM CAMPAIGNS

The abuse scenario assumes two entities – client and C2 domain. The client unwillingly initiates one or multiple DoH requests to the C2 domain – the response contains a JavaScript code or pieces of code. The client then executes the code and performs actions commanded by the C2 server.

This scenario is observed mainly in redirection use-cases, often triggered by illegal advertisements. Its usage was detected in e-mail spam campaigns; however, the same scripts can be found even on websites. All of the detected scripts utilized the same principle as in Section V-B — C2 domain queried via the JSON DoH API of Google resolver, therefore it falls into a *C2 Access and Communication* category of

[17]Top Level Domain

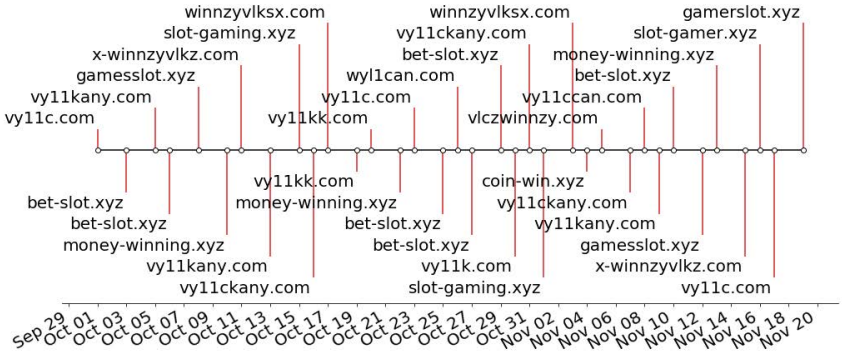

FIGURE 4. Responded landing domains in time for selected C2 between September 29th and November 20th in 2020. For data extraction we used Security Trails Passive DNS System.

```
//base64 encoded query to C2 domain
query = b64decode('aHR0...90LnR2')
function redirector(){
   // b64encoded DoH query to C2 domain
   resp = fetch(query)
   command = json.parse(resp).txtContent
   // eval(window.location.redirect(
   //https://identification.some.domain))
   eval(command)
}
//The redirector can be directly called
redirector();
//It can utilize some API
//Example with googles OAuth API
<script src="https://accounts.google.com/o/oauth/
revoke?callback=redirector()">
</script>

//Or it can be triggerd by users' action
<button onClick="redirector()"></button>
```

Listing 4. DoH redirector in advertisements.

DoH abuse described in Section IV-A. However, contrary to Scenario 2, these scripts did not use JavaScript Service Worker API; instead, they fetched JavaScript source code from the C2 channel and executed it.

The samples we observed on websites received redirection JavaScript code to illegally operated web pages. The C2 communication was fetched usually right after the load or by some action such as a button click. In the case of e-mail spam campaigns, an HTML document is delivered as an attachment (or as a MIME[18] part) and requires the mail client to open it. The pseudocode of the malicious scripts is shown in Listing 4. At first, the DoH TXT query to the attacker's C2 domain is performed. The attacker domain is usually encoded as a base64 string and hardcoded in the script. The DoH request can be executed directly within scripts. We also observed utilization of public API (such as Google OAuth API), in which case the malicious code is passed as a callback function.

In the observed cases, the answer always contained the redirection script with a landing URL wrapped inside a code utilizing JavaScript *window* API. The JavaScript interpreter then executed the code and performed the redirection. Even though we observed its use only in redirection use-cases, passing a JavaScript code from the C2 domain gives the attacker immense flexibility to run almost any command. Such practice can make phishing and cross-site scripting attacks more resistant because exploiting public DoH resolvers hides them to the network traffic analysis systems, which could trigger an alarm if the JavaScript code was downloaded directly by HTTPS from a potentially suspicious domain.

[18]Multipurpose Internet Mail Extensions

TABLE 7. Summary of presented abuse scenarios characteristics. The abbrevation of abuse category stands for: **CMC** — Covert Multipurpose Channels (Section IV-B), **C2** — C2 Access and Communication (Section IV-A).

	Scen. 1	Scen. 2	Scen. 3
Requires specialized client	✓		
DoH as hidden channel to bypass DNS	✓	✓	✓
DoH as C2 channel		✓	✓
Getting malicious code from C2			✓
Targets only Web Browsers		✓	
Website closure detection		✓	
Abuse category	CMC	C2	C2

SUMMARY OF DESCRIBED ABUSE SCENARIOS

The described scenarios represent working examples of mechanisms built above DoH that 1) have been observed by our malware laboratory, 2) we are not aware of their description in any previous academic study, and 3) can be very easily used for any malicious activity. Each scenario misuse DoH in a different way, Scenario 1 uses DoH to bypass restricted DNS, Scenario 2 detects DNS tampering and website closure, and then performs redirection. Moreover, scenario 2 uses DoH as a C2 to recognize valid web pages from the block page. Scenario 3 also uses DoH as a C2; however, it uses DoH for obtaining malicious code. The differences between scenarios are also shown in Table 7.

VI. FUTURE RESEARCH CHALLENGES

Based on our experience in network security, encrypted traffic analysis and survey of related works, we have identified several interesting open research challenges related to DoH. The following paragraphs explain them.

DoH Blocking/Filtering The goal of this research area is to identify DoH communication and block it timely. The possibility to identify and stop DoH allows prevention

of connection to non-permitted and untrustworthy DoH providers — when defined security policies permit only a particular one. Currently, it is not feasible to block DoH by standard firewall mechanisms based on IP addresses and ports because 1) IP addresses might change or any new DoH provider can be established; additionally, malware can easily use IP that is not well-known, 2) DoH shares the same TCP port number with other legitimate HTTPS traffic. Current methods [31], [32], [35] can detect only long DoH connections from web browsers; thus, single query DoH connections remain undetected.

Detection of Legitimate/Illegal Use In some countries, some content might be prohibited. Even though this scope of research can be misused for censorship or propaganda, some cases are still globally assumed to be harmful to society, such as drugs, child pornography, gambling, or illegal weapons. The point is that it is challenging to recognize specific topics of content inside encrypted traffic. A potential solution could be the adaptation of website fingerprinting approaches to DoH.

Detection of Malicious Use Besides illegal use mentioned above, adversary/malicious use of DoH by malicious software also presents a threat, which needs to be addressed by security research. Research studies concerning malicious use focused mainly on the detection of data exfiltration (see Table 4), leaving other misuses, such as C2 communication, undetected.

Detection of System Bypassing In this paper, we described some principles of how existing tools bypass the standard way of communication (e.g., JavaScript can perform DoH requests directly to lookup the final destination of some content). Generally, detecting such behavior is non-trivial due to possible obfuscation of the source codes. Therefore, it is a challenge to discover such a mechanism either in the web content (static analysis is not enough and probably dynamic analysis must be used) inside a web browser or at the network level based on behavioral analysis of standard and anomalous characteristics of the traffic. It is quite doubtful whether a JavaScript or any web browser plugin should be permitted to perform domain name resolving without any auditing; however, detecting such behavior would help identify serious security policy violations.

For completeness, any network-level anomaly detection brings common challenges like missing ground truth and handling false positives issues.

VII. CONCLUSION

DNS over HTTPS is a new rapidly-disseminating technology, which is becoming a popular alternative as the domain resolution mechanism instead of unencrypted DNS. Even though DoH improves users' privacy, our study highlighted that it also provides opportunities to threat actors. DoH will likely be the new reality for everyone who wants to resolve domain names to IP addresses because it is getting enabled by default. Consequently, the rise of legitimate DoH traffic amplifies the risk of hidden malicious activities inside DoH traffic, which currently cannot be detected easily.

Extensive DoH threat monitoring in our malware laboratory and manual analysis of code samples has led us to find DoH abuse by websites, which no study has previously described. The three DoH abuse scenarios are described on real-world examples: 1) a client can resolve a domain that would be otherwise blocked due to copyright infringement law, 2) DoH is used in website redirections in such a way that the user is transparently redirected to the latest working locations which are obtained privately via DoH, and 3) clicking on links distributed in malvertising and spam campaigns can trigger DoH queries that return and execute arbitrary JavaScript code.

Moreover, we analyzed multiple public sources (such as, Github, VirusTotal) and malware analysis blogs to summarize the knowledge about known DoH abuse by malware and proof-of-concept codes. On top of that, we taxonomize known DoH abuse into three categories: 1) Command and Control, 2) Covert Multipurpose Channel, which is mainly used for exfiltration, and 3) Unaware Usage.

User privacy is one of the essential priorities in modern society; however, in this paper, we pointed out and summarized the danger that DoH enables. Since DoH is encrypted, it prevents traditional security analysis and detection. It is highly expected that new generations of malware will exploit DoH for command and control communication, malware distribution, or data exfiltration. Moreover, with the broader deployment of DoH proxies, even malware that relies on the traditional DNS can unintentionally exploit improved privacy and avoid detection.

To better understand the benefits and risks of DoH, we call for more profound research on DoH abuse on the web. As a starting point for the network security research community, we have listed some open research challenges that can help to accelerate it.

REFERENCES

[1] P. E. Hoffman and P. McManus, *DNS Queries Over HTTPS (DoH)*, document 8484, Oct. 2018, p. 21. [Online]. Available: https://www.rfc-editor.org/info/rfc8484, doi: 10.17487/RFC8484.

[2] P. Mockapetris, "Domain names—Implementation and specification," Standard RFC 1035, 1983, pp. 1–55. [Online]. Available: https://www.rfc-editor.org/rfc/rfc1035.txt

[3] E. Brumaghin and C. Grady, "Covert channels and poor decisions: The tale of dnsmessenger," *Talos Intell. Group Comprehensive Threat Intell.*, Mar. 2017. [Online]. Available: https://blog.talosintelligence.com/2017/03/dnsmessenger.html

[4] C. Cimpanu. (Dec. 2020). *Here's How to Enable DoH in Each Browser, ISPs be Damned*. [Online]. Available: https://www.zdnet.com/article/dns-over-https-willeventually-roll-out-in-all-major-browsers-despite-isp-opposition/

[5] P. E. Hoffman, "Representing DNS messages in JSON," Standard RFC 8427, 2018. [Online]. Available: https://rfc-editor.org/rfc/rfc8427.txt

[6] S. García, K. Hynek, D. Vekshin, T. Čejka, and A. Wasicek, "Large scale measurement on the adoption of encrypted DNS," 2021, *arXiv:2107.04436*.

[7] K. Hynek and T. Cejka, "Privacy illusion: Beware of unpadded DoH," in *Proc. 11th IEEE Annu. Inf. Technol., Electron. Mobile Commun. Conf. (IEMCON)*, Nov. 2020, pp. 0621–0628.

[8] 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," in *Proc. 47th Res. Conf. Commun., Inf. Internet Policy*, Jul. 2019, pp. 1–9.

[9] I. N. Bozkurt, A. Aguirre, B. Chandrasekaran, P. B. Godfrey, G. Laughlin, B. Maggs, and A. Singla, "Why is the internet so slow?!" in *Passive and Active Measurement*, M. A. Kaafar, S. Uhlig, and J. Amann, Eds. Cham, Switzerland: Springer, 2017, pp. 173–187.

[10] P. McManus. (Aug. 2018). *Firefox Nightly Secure DNS Experimental Results — Firefox Nightly News*. [Online]. Available: https://blog.nightly.mozilla.org/2018/08/28/firefox-nightly-secure-dns-experimental-results/

[11] T. Böttger, F. Cuadrado, G. Antichi, E. L. Fernandes, G. Tyson, I. Castro, and S. Uhlig, "An empirical study of the cost of DNS-over-HTTPS," in *Proc. Internet Meas. Conf.* New York, NY, USA: ACM, Oct. 2019, pp. 15–21, doi: 10.1145/3355369.3355575.

[12] A. Hounsel, P. Schmitt, K. Borgolte, and N. Feamster, "Can encrypted DNS be fast?" in *Passive and Active Measurement*, O. Hohlfeld, A. Lutu, and D. Levin, Eds. Cham, Switzerland: Springer, 2021, pp. 444–459.

[13] K. Jerabek, O. Rysavy, and I. Burgetova, "Measurement and characterization of DNS over HTTPS traffic," 2022, *arXiv:2204.03975*.

[14] R. Chhabra, P. Murley, D. Kumar, M. Bailey, and G. Wang, "Measuring DNS-over-HTTPS performance around the world," in *Proc. 21st ACM Internet Meas. Conf.* New York, NY, USA: ACM, 2021, pp. 351–365, doi: 10.1145/3487552.3487849.

[15] A. Hounsel, K. Borgolte, P. Schmitt, J. Holland, and N. Feamster, "Comparing the effects of DNS, DoT, and DoH on web performance," in *Proc. Web Conf.* New York, NY, USA: ACM, 2020, pp. 562–572, doi: 10.1145/3366423.3380139.

[16] E. S. Mbewe and J. Chavula, "On QoE impact of DoH and DoT in Africa: Why a user's DNS choice matters," in *Towards New E-Infrastructure and E-Services for Developing Countries*, R. Zitouni, A. Phokeer, J. Chavula, A. Elmokashfi, A. Gueye, and N. Benamar, Eds. Cham, Switzerland: Springer, 2021, pp. 289–304.

[17] T. Jensen. (May 2020). *Windows Insiders Can Now Test DNS Over HTTPS*. [Online]. Available: https://techcommunity.microsoft.com/t5/networking-blog/windows-insiders-can-now-test-dns-over-https/ba-p/1381282

[18] C. Deccio and J. Davis, "DNS privacy in practice and preparation," in *Proc. 15th Int. Conf. Emerg. Netw. Exp. Technol.* New York, NY, USA: ACM, Dec. 2019, pp. 138–143, doi: 10.1145/3359989.3365435.

[19] T. Wicinski, *DNS privacy considerations*, document 9076, Jul. 2021. [Online]. Available: https://www.rfc-editor.org/info/rfc9076, doi: 10.17487/RFC9076.

[20] H. Shulman, "Pretty bad privacy: Pitfalls of DNS encryption," in *Proc. 13th Workshop Privacy Electron. Soc.*, Nov. 2014, pp. 191–200.

[21] A. Mayrhofer, "The EDNS(0) padding option," Standard RFC 7830, 2016.

[22] S. Siby, M. Juarez, C. Diaz, N. Vallina-Rodriguez, and C. Troncoso, "Encrypted DNS ⇒ privacy? A traffic analysis perspective," Dec. 2020, *arXiv:1906.09682*.

[23] J. Bushart and C. Rossow, "Padding Ain't enough: Assessing the privacy guarantees of encrypted DNS," 2019, *arXiv:1907.01317*.

[24] Q. Huang, D. Chang, and Z. Li, "A comprehensive study of DNS-over-HTTPS downgrade attack," in *Proc. 10th USENIX Workshop Free Open Commun. Internet (FOCI)*, 2020, pp. 1–8.

[25] S. Dickinson, D. K. Gillmor, and T. K. Reddy, "Usage profiles for DNS over TLS and DNS over DTLS," Standard RFC 8310, Mar. 2018. [Online]. Available: https://www.rfc-editor.org/info/rfc8310

[26] S. Singanamalla, S. Chunhapanya, M. Vavruša, T. Verma, P. Wu, M. Fayed, K. Heimerl, N. Sullivan, and C. Wood, "Oblivious DNS over HTTPS (ODoH): A practical privacy enhancement to DNS," 2020, *arXiv:2011.10121*.

[27] E. Kinnear, P. McManus, T. Pauly, T. Verma, and C. A. Wood, "Oblivious DNS Over HTTPS," Internet Engineering Task Force, Fremont, CA, USA, Tech. Rep. draft-pauly-dprive-oblivious-doh-11, Feb. 2022, p. 21. [Online]. Available: https://datatracker.ietf.org/doc/html/draft-pauly-dprive-oblivious-doh-11

[28] A. Fidler, B. Hubert, J. Livingood, J. Reid, and N. Leymann, "DNS over HTTPS (DoH) considerations for operator networks," Internet Engineering Task Force, Fremont, CA, USA, Tech. Rep., Mar. 2019. [Online]. Available: https://datatracker.ietf.org/doc/html/draft-reid-doh-operator-00

[29] K. Bumanglag and H. Kettani, "On the impact of DNS over HTTPS paradigm on cyber systems," in *Proc. 3rd Int. Conf. Inf. Comput. Technol. (ICICT)*, Mar. 2020, pp. 494–499.

[30] R. Badhwar, *Domain Name System (DNS) Security*. Cham, Switzerland: Springer, 2021, pp. 207–212, doi: 10.1007/978-3-030-75354-2_24.

[31] M. MontazeriShatoori, L. Davidson, G. Kaur, and A. H. Lashkari, "Detection of DoH tunnels using time-series classification of encrypted traffic," in *Proc. IEEE Int. Conf. Dependable, Autonomic Secure Comput., Int. Conf. Pervasive Intell. Comput., Int. Conf. Cloud Big Data Comput., Int. Conf. Cyber Sci. Technol. Congr. (DASC/PiCom/CBDCom/CyberSciTech)*, Aug. 2020, pp. 63–70.

[32] D. Vekshin, K. Hynek, and T. Cejka, "DoH insight: Detecting DNS over HTTPS by machine learning," in *Proc. 15th Int. Conf. Avail., Rel. Secur.* New York, NY, USA: ACM, 2020, doi: 10.1145/3407023.3409192.

[33] Y. M. Banadaki, "Detecting malicious DNS over HTTPS traffic in domain name system using machine learning classifiers," *J. Comput. Sci. Appl.*, vol. 8, no. 2, pp. 46–55, Aug. 2020.

[34] S. K. Singh and P. K. Roy, "Detecting malicious DNS over HTTPS traffic using machine learning," in *Proc. Int. Conf. Innov. Intell. Informat., Comput. Technol. (3ICT)*, Dec. 2020, pp. 1–6.

[35] J. Wu, Y. Zhu, B. Li, Q. Liu, and B. Fang, "Peek inside the encrypted world: Autoencoder-based detection of DoH resolvers," in *Proc. IEEE 20th Int. Conf. Trust, Secur. Privacy Comput. Commun. (TrustCom)*, Oct. 2021, pp. 783–790.

[36] L. F. G. Casanova and P.-C. Lin, "Generalized classification of DNS over HTTPS traffic with deep learning," in *Proc. Asia–Pacific Signal Inf. Process. Assoc. Annu. Summit Conf. (APSIPA ASC)*, Dec. 2021, pp. 1903–1907.

[37] L. Csikor, H. Singh, M. S. Kang, and D. M. Divakaran, "Privacy of DNS-over-HTTPS: Requiem for a dream?" in *Proc. IEEE Eur. Symp. Secur. Privacy (EuroS&P)*, Sep. 2021, pp. 252–271.

[38] C. Kwan, P. Janiszewski, S. Qiu, C. Wang, and C. Bocovich, "Exploring simple detection techniques for DNS-over-HTTPS tunnels," in *Proc. ACM SIGCOMM Workshop Free Open Commun. Internet*, Aug. 2021, pp. 37–42.

[39] S. Ding, D. Zhang, J. Ge, X. Yuan, and X. Du, "Encrypt DNS traffic: Automated feature learning method for detecting DNS tunnels," in *Proc. IEEE Int. Conf. Parallel Distrib. Process. Appl., Big Data Cloud Comput., Sustain. Comput. Commun., Social Comput. Netw. (ISPA/BDCloud/SocialCom/SustainCom)*, Sep. 2021, pp. 352–359.

[40] M. Behnke, N. Briner, D. Cullen, K. Schwerdtfeger, J. Warren, R. Basnet, and T. Doleck, "Feature engineering and machine learning model comparison for malicious activity detection in the DNS-over-HTTPS protocol," *IEEE Access*, vol. 9, pp. 129902–129916, 2021.

[41] R. Alenezi and S. A. Ludwig, "Classifying DNS tunneling tools for malicious DoH traffic," in *Proc. IEEE Symp. Ser. Comput. Intell. (SSCI)*, Dec. 2021, pp. 1–9.

[42] T. Zebin, S. Rezvy, and Y. Luo, "An explainable AI-based intrusion detection system for DNS over HTTPS (DoH) attacks," Tech. Rep., Jan. 2022. [Online]. Available: https://www.techrxiv.org/articles/preprint/An_Explainable_AI-based_Intrusion_Detection_System_for_DNS_over_HTTPS_DoH_Attacks/17696972, doi: 10.36227/techrxiv.17696972.v1.

[43] M. Zhan, Y. Li, G. Yu, B. Li, and W. Wang, "Detecting DNS over HTTPS based data exfiltration," *Comput. Netw.*, vol. 209, May 2022, Art. no. 108919. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1389128622001104

[44] C. J. Dietrich, C. Rossow, F. C. Freiling, H. Bos, M. V. Steen, and N. Pohlmann, "On botnets that use DNS for command and control," in *Proc. 7th Eur. Conf. Comput. Netw. Defense*, Sep. 2011, pp. 9–16.

[45] K. Xu, P. Butler, S. Saha, and D. Yao, "DNS for massive-scale command and control," *IEEE Trans. Dependable Secure Comput.*, vol. 10, no. 3, pp. 143–153, May 2013.

[46] H. Binsalleeh, A. M. Kara, A. Youssef, and M. Debbabi, "Characterization of covert channels in DNS," in *Proc. 6th Int. Conf. New Technol., Mobility Secur. (NTMS)*, Mar. 2014, pp. 1–5.

[47] W. Ellens, P. Żuraniewski, A. Sperotto, H. Schotanus, M. Mandjes, and E. Meeuwissen, "Flow-based detection of DNS tunnels," in *Emerging Management Mechanisms for the Future Internet*, G. Doyen, M. Waldburger, P. Čeleda, A. Sperotto, and B. Stiller, Eds. Berlin, Germany: Springer, 2013, pp. 124–135.

[48] C. Qi, X. Chen, C. Xu, J. Shi, and P. Liu, "A bigram based real time DNS tunnel detection approach," *Proc. Comput. Sci.*, vol. 17, pp. 852–860, Jan. 2013, doi: 10.1016/j.procs.2013.05.109.

[49] T. Cejka, Z. Rosa, and H. Kubatova, "Stream-wise detection of surreptitious traffic over DNS," in *Proc. IEEE 19th Int. Workshop Comput. Aided Modeling Design Commun. Links Netw. (CAMAD)*, Athens, Greece, Dec. 2014, pp. 300–304.

[50] S. Sheridan and A. Keane, "Detection of DNS based covert channels," in *Proc. Eur. Conf. Cyber Warfare Secur.* Reading, U.K.: Academic Conf. Int. Ltd., 2015, p. 267.

[51] T. A. Peña, "A deep learning approach to detecting covert channels in the domain name system," Ph.D. dissertation, Capitol Technology University, Laurel, MD, USA, 2020.
[52] (Dec. 2020). *Cisco 2016 Annual Cybersecurity Report*. [Online]. Available: http://mkto.cisco.com/rs/564-whv-323/images/cisco-asr-2016.pdf
[53] A. K. Sood and S. Zeadally, "A taxonomy of domain-generation algorithms," *IEEE Secur. Privacy*, vol. 14, no. 4, pp. 46–53, Jul./Aug. 2016.
[54] J. Nazario and T. Holz, "As the net Churns: Fast-flux botnet observations," in *Proc. 3rd Int. Conf. Malicious Unwanted Softw. (MALWARE)*, Oct. 2008, pp. 24–31.
[55] T. Alex. (2019). *An Analysis of Godlua Backdoor*. [Online]. Available: https://blog.netlab.360.com/an-analysis-of-godlua-backdoor-en/
[56] R. Vaughn and G. Evron. (Mar. 2006). *DNS Amplification Attacks*. [Online]. Available: http://web.archive.org/web/20110717124634/
[57] The Proofpoint Threat Insight. (Dec. 2020). *PsiXBot Now Using Google DNS Over HTTPS and Possible New Sexploitation Module: Proofpoint US*. [Online]. Available: https://www.proofpoint.com/us/threat-insight/post/psixbot-now-using-google-dns-over-https-and-possible-new-sexploitation-module
[58] L. Winter. (Dec. 2021). *FluBot Malware–All You Need to Know &; to Act Now*. [Online]. Available: https://www.threatmark.com/flubot-banking-malware/
[59] J. Hammond. (Sep. 2020). *Hiding in Plain Sight || Part 2*. [Online]. Available: https://blog.huntresslabs.com/hiding-in-plain-sight-part-2-dfec817c036f
[60] (Jul. 2021). *Pink, a Botnet That Competed With the Vendor to Control the Massive Infected Devices*. [Online]. Available: https://blog.netlab.360.com/pink-en/
[61] A. Threat. (Dec. 2020). *Illicit Cryptomining Threat actor Rocke Changes Tactics, Now More Difficult to Detect*. [Online]. Available: https://www.anomali.com/blog/illicit-cryptomining-threat-actor-rocke-changes-tactics-now-more-difficult-to-detect
[62] J. Leon. (Dec. 2020) *Waiting for Godoh*. [Online]. Available: https://sensepost.com/blog/2018/waiting-for-godoh/
[63] SpiderLabs. (Dec. 2020). *Spiderlabs/DoHC2*. [Online]. Available: https://github.com/SpiderLabs/DoHC2
[64] (Dec. 2020). *Arno0x/Dnsexfiltrator*. [Online]. Available: https://github.com/Arno0x/DNSExfiltrator
[65] (Dec. 2020). *In & Out—The Network Data Exfiltration Techniques Training*. [Online]. Available: https://44con.com/44con-training/inout-the-network-data-exfiltration-techniques-training/
[66] D. Stevens. (Dec. 2020). *Downloading Executables Over DNS: Capture Files*. [Online]. Available: https://blog.didierstevens.com/2019/08/07/downloading-executables-over-dns-capture-files/
[67] R. Radu and M. Hausding, "Consolidation in the DNS resolver market—How much, how fast, how dangerous?" *J. Cyber Policy*, vol. 5, no. 1, pp. 46–64, Jan. 2020, doi: 10.1080/23738871.2020.1722191.
[68] SensePost Ethical Hacking Team. (Dec. 2020). *Sensepost/Godoh*. [Online]. Available: https://github.com/sensepost/godoh
[69] D. Fifield. (2020). *Dnstt — DoH- and DoT-capable DNS tunnel*. [Online]. Available: https://www.bamsoftware.com/software/dnstt/index.html
[70] C. Cimpanu. (Dec. 2020). *Iranian Hacker Group Becomes First Known APT to Weaponize DNS-Over-HTTPS (DoH)*. [Online]. Available: https://www.zdnet.com/article/iranian-hacker-group-becomes-first-known-apt-to-weaponize-dns-over-https-doh/
[71] E. Ekman, B. Andersson, and A. Bezemer. (Dec. 2020). *Yarrick/Iodine*. [Online]. Available: https://github.com/yarrick/iodine/tree/iodine-0.7
[72] T. Pietraszek. (Dec. 2020). *DNScat*. [Online]. Available: http://tadek.pietraszek.org/projects/DNScat/
[73] L. Nussbaum. *Dec. 2020. Lnussbaum/Tuns*. [Online]. Available: https://github.com/lnussbaum/tuns
[74] S. Neef, "Performance of iodine over DNS-over-HTTPS," Tech. Rep., Feb. 2019. [Online]. Available: https://0day.work/performance-of-iodine-over-dns-over-https/
[75] J. L. Hall, M. D. Aaron, S. Adams, B. Jones, and N. Feamster, "A survey of worldwide censorship techniques," Internet Engineering Task Force, Fremont, CA, USA, Tech. Rep., Mar. 2019. [Online]. Available: https://datatracker.ietf.org/doc/html/draft-hall-censorship-tech-07
[76] R. Deibert, J. Palfrey, R. Rohozinski, and J. Zittrain, *Access Denied: The Practice and Policy of Global Internet Filtering*. Cambridge, MA, USA: MIT Press, 2008.
[77] E. Rubin and N. Tucker, "TechNation: Pitango launches seventh venture fund with projected $175 million," Tech. Rep., Dec. 2016. [Online]. Available: https://www.haaretz.com/israel-news/business/technation-1.5474071
[78] P. Levis, "The collateral damage of internet censorship by DNS injection," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 42, no. 3, pp. 21–27, Jun. 2012.
[79] V. Marionneau, "Gambling in Russia: Policies, markets, and research," *Int. J. Russian Stud.*, vol. 9, no. 2, pp. 122–134, 2020.
[80] D. Fifield, C. Lan, R. Hynes, P. Wegmann, and V. Paxson, "Blocking-resistant communication through domain fronting," *Proc. Privacy Enhancing Technol.*, vol. 2015, no. 2, pp. 46–64, Jun. 2015. [Online]. Available: https://content.sciendo.com/view/journals/popets/2015/2/article-p46.xml
[81] A. Johnson, "Domain fronting: Making backdoor access look like Google requests," Tech. Rep., 2018. [Online]. Available: https://www.cs.tufts.edu/comp/116/archive/spring2018/ajohnson.pdf

KAREL HYNEK received the bachelor's degree. He is currently pursuing the Ph.D. degree in network security and network monitoring with the Faculty of Information Technology, Czech Technical University in Prague. He has worked on several national and international research projects related to security. He is a Key Member with the Network Traffic Monitoring Research Team, Faculty of Information Technology.

DMITRII VEKSHIN is currently pursuing the master's degree with the Faculty of Information Technology, Czech Technical University in Prague. He is a member with the Network Traffic Monitoring Research Team, Faculty of Information Technology. He is also a Researcher at Avast, working in the field of machine learning and network security. His recent research interests include encrypted traffic monitoring and particularly encrypted DNS.

JAN LUXEMBURK is currently pursuing the Ph.D. degree with the Faculty of Information Technology, Czech Technical University in Prague, with a focus on detecting threats in encrypted network traffic. He is also working as a Researcher at CESNET a.l.e., the operator of the Czech national research and educational networks. His research interest includes brute-force attacks in encrypted HTTPS communications.

TOMAS CEJKA received the Ph.D. degree in network security, more specifically network traffic monitoring and analysis area. He is currently an Assistant Professor at the Faculty of Information Technology, Czech Technical University in Prague. He teaches the network security course, and supervises a research group of bachelor's/master's/Ph.D. students. Additionally, he works as a Researcher and a Team Leader at CESNET a.l.e., the operator of the Czech national research and education networks.

ARMIN WASICEK (Senior Member, IEEE) received the Ph.D. and M.Sc. degrees from Technical University Vienna, Austria. He was a Marie Curie Fellow at the University of California at Berkeley. He is currently a Research Manager at Avast, working on network security for smart home networks. His research interests include machine learning, security, and the Internet of Things.