A Survey on DNS Encryption: Current Development, Malware Misuse, and Inference Techniques

MINZHAO LYU, University of New South Wales, Australia
HASSAN HABIBI GHARAKHEILI, University of New South Wales, Australia
VIJAY SIVARAMAN, University of New South Wales, Australia

The domain name system (DNS) that maps alphabetic names to numeric Internet Protocol (IP) addresses plays a foundational role in Internet communications. By default, DNS queries and responses are exchanged in unencrypted plaintext, and hence, can be read and/or hijacked by third parties. To protect user privacy, the networking community has proposed standard encryption technologies such as DNS over TLS (DoT), DNS over HTTPS (DoH), and DNS over QUIC (DoQ) for DNS communications, enabling clients to perform secure and private domain name lookups. We survey the DNS encryption literature published from 2016 to 2021, focusing on its current landscape and how it is misused by malware, and highlighting the existing techniques developed to make inferences from encrypted DNS traffic. First, we provide an overview of various standards developed in the space of DNS encryption and their adoption status, performance, benefits, and security issues. Second, we highlight ways that various malware families can exploit DNS encryption to their advantage for botnet communications and/or data exfiltration. Third, we discuss existing inference methods for profiling normal patterns and/or detecting malicious encrypted DNS traffic. Several directions are presented to motivate future research in enhancing the performance and security of DNS encryption.

CCS Concepts: • **Networks → Application layer protocols**; *Naming and addressing*; • **Security and privacy → Security protocols**; **Malware and its mitigation**.

Additional Key Words and Phrases: DNS encryption, DoT, DoH, DoQ, malware communitactions

ACM Reference Format:
Minzhao Lyu, Hassan Habibi Gharakheili, and Vijay Sivaraman. 2022. A Survey on DNS Encryption: Current Development, Malware Misuse, and Inference Techniques. 1, 1 (July 2022), 27 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

The domain name system (DNS) protocol [75] takes the responsibility of converting human-readable domain names into machine-friendly IP addresses and vice versa, which is critical to the Internet communications today. We start by illustrating the basic operation of DNS in Fig. 1: a client embeds its question (typically for domain names) in DNS queries and sends them to DNS resolvers for lookups of the answers (step ①). The resolver then performs recursive lookups of the questioned domain name by successively/recursively querying a root server, top-level-domain (TLD) server, and authoritative name server (steps ②, ③, and ④ in Fig. 1, respectively).

The DNS ecosystem presents various types of security and privacy risks that could be exploited by malicious actors. For example, DNS resolvers and name servers often become attractive targets of denial-of-service (DoS) attacks that

Authors' addresses: Minzhao Lyu, University of New South Wales, Sydney, Australia, minzhao.lyu@unsw.edu.au; Hassan Habibi Gharakheili, University of New South Wales, Sydney, Australia, h.habibi@unsw.edu.au; Vijay Sivaraman, University of New South Wales, Sydney, Australia, vijay@unsw.edu.au.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

© 2022 Association for Computing Machinery.
Manuscript submitted to ACM

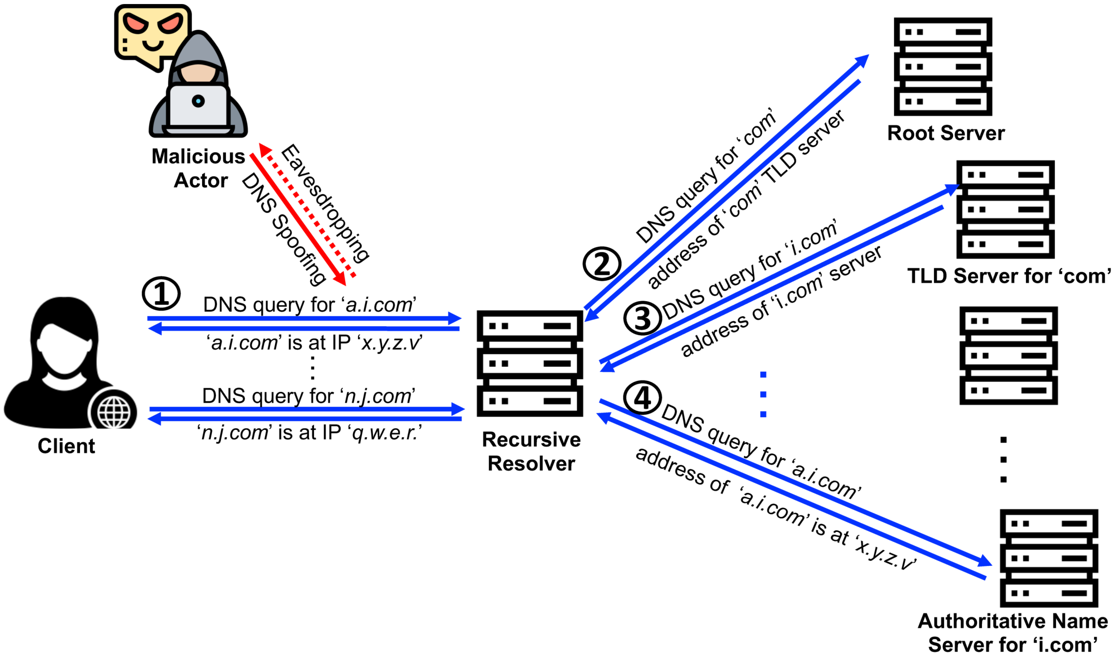
Malicious Actor
Eavesdropping
DNS Spoofing
Client
① DNS query for 'a.i.com'
'a.i.com' is at IP 'x.y.z.v'
DNS query for 'n.j.com'
'n.j.com' is at IP 'q.w.e.r.'
Recursive Resolver
② DNS query for 'com'
address of 'com' TLD server
③ DNS query for 'i.com'
address of 'i.com' server
④ DNS query for 'a.i.com'
address of 'a.i.com' is at 'x.y.z.v'
Root Server
TLD Server for 'com'
Authoritative Name Server for 'i.com'

Fig. 1. A visual example of DNS lookup activities from a client and the potential threats introduced by malicious actors.

slow down or even paralyze their services. For countermeasures, there exist specialized security systems and appliances that are able to detect and mitigate DoS attacks on DNS infrastructures included [16, 70]. It is also a common practice for attackers to mislead clients by fake responses (DNS spoofing and/or hijacking), directing them to malicious servers. To tackle this problem, DNSSEC [13] was proposed for protecting the data integrity of DNS. It provides cryptographic verification through digital signatures that can be used to validate the records delivered in a DNS response from the authoritative DNS server. Furthermore, as shown in Fig. 1, DNS queries and responses are communicated in plaintext via UDP transport-layer protocol which offers various benefits such as low computational overheads, fast resolution, and ease of deployment and management [6]. However, malicious actors take advantage of unencrypted contents, putting user privacy at risk [114] or spoofing [15] DNS lookups between clients and resolvers. To address privacy concerns, encrypting DNS lookups (exchanged between client and resolver) has been developed and promoted, which is the focus of this survey.

Various techniques of DNS encryption have been developed, experimented with, and deployed across the Internet over the past few years to protect user privacy. Some early proposals such as DNSCurve [34] and DNSCrypt [66] were introduced in 2008 and 2011 [92], respectively. However, none of them became officially standardized, and hence are not widely deployed by the Internet community. To facilitate the adoption of DNS encryption, the Internet Engineering Task Force (IETF) has proposed a series of standards (RFCs) starting from 2016, relying upon existing secure/private protocols, including DNS over TLS (DoT) [53], DNS over HTTPS (DoH) [46], and DNS over QUIC (DoQ) [55]. Currently, the public adoption of encrypted DNS (compared with plaintext DNS) is still relatively low, but several major Internet technology and service providers such as Google, Cloudflare, Cisco, and Alibaba have launched their encrypted DNS resolvers [5], fueling the growth of encrypted DNS traffic on the Internet [39].

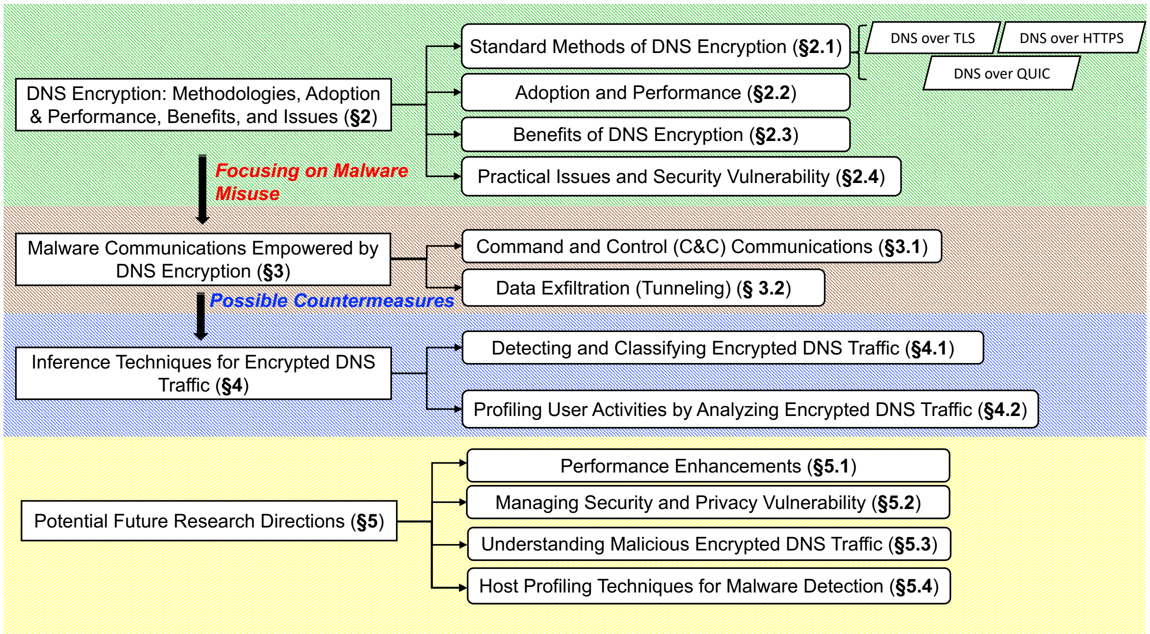
DNS Encryption: Methodologies, Adoption & Performance, Benefits, and Issues (§2)
Standard Methods of DNS Encryption (§2.1)
DNS over TLS
DNS over HTTPS
DNS over QUIC
Adoption and Performance (§2.2)
Benefits of DNS Encryption (§2.3)
Practical Issues and Security Vulnerability (§2.4)
Focusing on Malware Misuse
Malware Communications Empowered by DNS Encryption (§3)
Command and Control (C&C) Communications (§3.1)
Data Exfiltration (Tunneling) (§ 3.2)
Possible Countermeasures
Inference Techniques for Encrypted DNS Traffic (§4)
Detecting and Classifying Encrypted DNS Traffic (§4.1)
Profiling User Activities by Analyzing Encrypted DNS Traffic (§4.2)
Potential Future Research Directions (§5)
Performance Enhancements (§5.1)
Managing Security and Privacy Vulnerability (§5.2)
Understanding Malicious Encrypted DNS Traffic (§5.3)
Host Profiling Techniques for Malware Detection (§5.4)

Fig. 2. Key contributions and structure of this survey paper.

Regardless of the benefits brought by DNS encryption, some practical issues and security problems have been identified by the existing literature, impacting its public adoption. For example, encrypted resolution may not be feasible if encryption-enabled DNS resolvers fail to provide a valid certificate. Unfortunately, as reported in [69], lack of best practice encryption setups is not uncommon in today's public encrypted DNS resolvers. Moreover, inappropriately configured DNS resolvers are vulnerable to fallback (downgrade) attacks (*e.g.,* SSL stripping), forcing resolvers to return DNS responses in plaintext.

Among those practical issues, the misuse of DNS encryption by malware is one of the problems that has attracted attention from the security community. Cybercriminals can easily hide their identities and activities via encrypted DNS traffic to bypass legacy security tools and appliances. Many monitoring and detection methods today rely on DNS packet content inspection, which is highly effective in processing plaintext-based DNS communications. Existing literature has reported that malicious actors use encrypted DNS for command-and-control (C&C) communications and/or exfiltrating and tunneling data between malware-infected devices and cloud-based servers – further details will be discussed later in §3. To tackle the malicious usage of DNS encryption by malware, researchers have developed various analytical methods to detect encrypted DNS packets from network traffic and classify benign and anomalous streams. In addition, there are also research works that profile user behaviors by analyzing encrypted DNS traffic. Prior works collectively provide solid insights into challenges and opportunities, and motivate future researches in detecting and fingerprinting malware-infected devices that utilize encrypted DNS for stealthy communications.

This survey is the first to comprehensively review relevant literature (published from 2016 to 2021) on DNS encryption techniques, their opportunities, and risks. To guarantee that our survey has a full coverage of peer-reviewed papers in the domain of DNS encryption, we searched seven title keywords from eight major scientific digital libraries such

Table 1. List of references cited in each category of DNS encryption research.

	Category of topics covered by this survey	List of references
§2	Standardized DNS encryption methodologies	[2, 23, 27, 35, 38, 41, 46, 53, 55, 58, 64, 111]
§2	Adoption and performance	[7, 8, 14, 21, 22, 29, 33, 37, 39, 45, 47–49, 74, 84, 85, 94, 105, 110, 111, 113]
§2	Benefits of DNS encryption	[11, 24, 61, 82, 94, 113, 117]
§2	Practical issues and security vulnerability	[6, 9, 10, 20, 25, 30, 40, 50, 51, 54, 56, 57, 59, 63, 65, 69–71, 73, 80, 81, 88, 89, 91, 93, 97, 113]
§3	Malware misuse: C&C communications	[19, 24, 36, 43, 68, 83, 86, 87, 100, 102, 112, 115]
§3	Malware misuse: data exfiltration (or tunneling)	[1, 4, 9, 28, 44, 52, 79, 90, 103]
§4	Detecting and classifying encrypted DNS traffic	[17, 18, 26, 32, 67, 76, 87, 98, 107, 108, 108]
§4	Profiling user activities by analyzing encrypted DNS analysis	[51, 73, 77, 78, 95, 96, 106]

as IEEE Xplore and ACM Digital Library. Research papers we surveyed are from a wide range of academic journals, conferences, and workshops in general computer networking (*e.g.,* IEEE TNSM and Computer Networks), Internet measurement (*e.g.,* ACM IMC and PAM), and cybersecurity (*e.g.,* Computers & Security and USENIX Security). In addition, we incorporated relevant Internet standard documents (RFCs), technical reports from reputable organizations, and research papers that do not directly focus on encrypted DNS to cover certain key points around DNS encryption.

1.1 Contributions

The contributions of this survey paper can be summarized as follows.

- **First**, we outline the current development of DNS encryption, highlighting aspects like standard techniques (*i.e.,* DoT, DoH, and DoQ), the current status of their adoption across the Internet, performance analysis, benefits, practical issues, and security vulnerabilities identified by the current literature.
- **Second**, we discuss how DNS encryption can be misused by malware for purposes including command-and-control (C&C) communications and data exfiltration, with highlights on the (in)effectiveness of existing countermeasures originally developed for plaintext DNS when applied to encrypted DNS.
- **Third**, we survey the current analysis and inference methods for detecting encrypted DNS traffic from generic encrypted network streams (*e.g.,* HTTPS), classify malicious encrypted DNS communications, and fingerprint host profiles by analyzing encrypted DNS traffic. They provide strong references and motivations for future research in detecting malware-infected devices that exploit encrypted DNS protocol.
- **Last**, we identify four research directions in the field of DNS encryption yet to be investigated.

1.2 Roadmap

The organization of this paper is depicted in Fig. 2. We discuss the current development of DNS encryption in §2, malware communications leveraging DNS encryption in §3, and inference techniques for encrypted DNS traffic in §4. We also list the references cited in each of the above three sections in Table 1. Four future research directions are presented in §5. Related surveys on DNS security are discussed in §6. This survey is concluded in §7.

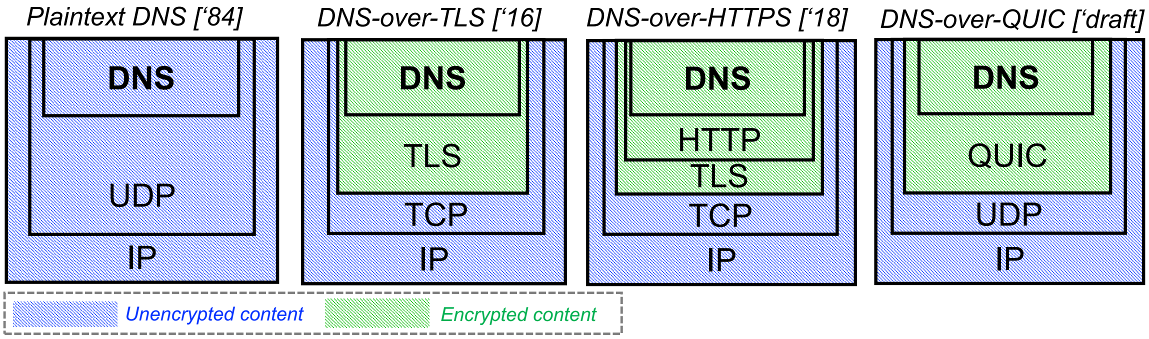

Fig. 3. The structure of packet layer (starting from IP layer) of plaintext DNS, encrypted DNS over TLS, encrypted DNS over HTTPS, and encrypted DNS over QUIC.

2 DNS ENCRYPTION: METHODOLOGIES, ADOPTION & PERFORMANCE, BENIFITS, AND ISSUES

Encrypting DNS queries and responses between clients and resolvers holds the promise of protecting user privacy against eavesdroppers and man-in-the-middle (MITM) attackers [111]. In this section, we review the development of DNS encryption by highlighting current standard techniques (§2.1), their adoption status and performance impacts (§2.2), benefits provided by DNS encryption (§2.3), and its open issues and security vulnerabilities (§2.4).

2.1 Standard Methods of DNS Encryption

In general, DNS encryption is achieved by encapsulating the content of queries and responses (between clients and resolvers) in an upper layer protocol that encrypts the packet content using available cryptographic techniques. To date, there are three standardized DNS encryption methods, each utilizing a specific upper-layer protocol, including DNS-over-TLS (DoT), DNS-over-HTTPS (DoH), and DNS-over-QUIC (DoQ), ordered by their time of proposal. As illustrated in Fig. 3, unlike the plaintext-based DNS that is directly embedded under the UDP transport layer, the three DNS encryption methods encapsulate DNS in their encryption-enabled layer, *i.e.,* TLS, TLS with HTTP (*i.e.,* HTTPS), and QUIC, respectively.

2.1.1 DNS over TLS (DoT). This is the first standard protocol proposed for DNS encryption that has its RFC [53] published in 2016. It uses a Transport Layer Security (TLS) layer under the TCP transport layer to encrypt DNS queries and responses. Before being applied to the DNS applications, TLS has proven its efficacy in encrypting popular network applications such as email (SMTP), hypertext (HTTP), and voice-over-IP (VoIP).

During lookups, the client first initiates a TCP connection to a designated DoT port `TCP/853` on its intended DoT-enabled resolver. Next, a TLS connection will be established via a typical TLS handshake process to exchange their cryptographic keys [35]. Upon successful establishment of the TLS session, the client is able to perform TLS-encrypted DNS lookups through the DoT port `TCP/853` on the resolver side. Depending on the configurations of clients and servers, the TLS connections may remain open for further DNS lookups, reducing latency (*i.e.,* preventing additional TCP/TLS handshakes for subsequent requests).

Although DoT is a viable approach for DNS encryption, it faces several practical challenges that may limit its usage. The first concern is that, as reported in [64], the port `TCP/853` dedicated to DoT services is currently not well recognized by the security community thus is likely to be blocked by firewall appliances. While the content of queries and responses is encrypted, eavesdroppers can obtain encrypted DNS packets relatively easier (compared to DoH and

DoT where a mix of traffic is exchanged) by filtering TCP/853 and performing statistical analysis (*e.g.,* on packet sizes) to infer their embedded contents (will be discussed in §4). Another shortcoming of DoT is that it requires application developers and hardware manufacturers to support the protocol – if not supported, users may still go unprotected. Note that DoT is less resilient to packet losses [99] (suffers from head-of-line blocking) compared to its counterparts, DoH and DoQ (discussed next). In addition to this technical shortcoming, the timing of when DoT was proposed seems relatively early (compared to DoH, for example) – it was introduced when encryption was not very commonplace. The above mentioned facts have negatively impacted the public adoption of DoT protocol.

2.1.2 DNS over HTTPS (DoH). DoH [46] has been standardized in 2018 to address the practical difficulties of DoT discussed above. Anyone using a supported web browser automatically benefits from encrypted DNS. It utilizes the widely-used HTTPS protocol (*i.e.,* TLS with HTTP as depicted in Fig. 3) to encapsulate DNS contents, delivered via the service port `TCP/443` so that existing security measures would not hinder its transmission. DoH is also more acceptable on the client-side as HTTPS is the default (or the only enabled) hypertext protocol used by major browsers. From a privacy viewpoint, DoH is more preferred than DoT since DNS traffic is mixed with other applications (*e.g.,* Web) exchanged over HTTPS. In addition, any future performance and security enhancements for HTTPS would also benefit DoH. Given the above practical benefits, DoH has become the most popular DNS encryption method used by the current Internet industry [38].

2.1.3 DNS over QUIC (DoQ). The transport-layer protocol of both DoT and DoH is TCP. Considering TCP and TLS handshakes required for connection establishment plus the TCP acknowledgement mechanism, a DNS request of DoT and/or DoH can take longer (compared to plaintext UDP-based) to get responded. Such overhead is often non-negligible Therefore, DoQ [55] was first proposed in 2017 by Google as an Internet draft (not officially finalized as an RFC yet) to further improve the performance of encrypted DNS. It leverages QUIC [58] protocol (built on the top of UDP transport layer) that enables faster and lighter encrypted communications by zero-RTT handshakes [41] and multiplexing data streams. As reported by G. Carlucci *et al.* [27], QUIC empirically outperforms its competitor HTTPS in response quality metrics like page load times.

Despite its performance merits, similar to DoT, DoQ uses dedicated service ports (`UDP/784` and `UDP/8853`) that are not well-recognized by security systems at present time, therefore, its traffic might get blocked (as the default option) by firewalls during transmission.

2.2 Adoption and Performance of DNS Encryption

Having understood the current DNS encryption techniques and protocols, we now discuss their state of adoption across the Internet as well as their performance reported by the data networking community.

2.2.1 Current State of Public Adoption. As described in §2.1, DNS encryption is applied for the queries and responses between clients and resolvers, and hence, the public adoption highly depends on the technical supports from two key stakeholders, *i.e.,* public resolvers and user applications (*e.g.,* browsers and operating systems).

Public Resolvers: A few public resolvers, mostly operated by major cloud providers such as Google and Cloudflare, start to support DNS encryption since 2019 [105], followed by many other providers such as AdGuard, Alibaba, Cisco, Comcast, and Quad9. According to two technical reports [7, 8], there are at least seven DoT-enabled and 62 DoH-enabled public resolvers on the Internet. AdGuard announced [14] the support of DoQ on their public resolvers since December 2020, though this protocol is still in its experimental stage.

Table 2. DNS encryption options provided by user applications.

Option	Description
Off	Plaintext without attempting to apply encryption for DNS communications.
Opportunistic/Automatic [23]	Apply encryption if possible by both client and server; otherwise, plaintext DNS (if no encryption protocol is available/possible).
Strict	Apply specified encrypted DNS protocols.

The public adoption of DNS encryption holds a growing promise, evidenced by its increasing Internet-wide traffic volumes [39]. However, it is dominated by a small number of major service providers, raising concerns of data monopolization [110]. Also, there are oppositions, like the UK Internet Services Providers' Association (ISPA) [84], who argue that DNS encryption can bypass existing filtering obligations, thus undermine the Internet safety standards.

User Applications: DNS encryption is increasingly supported by a variety of end-user platforms and applications including operating systems for mobile and desktop devices, Internet broswers, and even IoT devices [45]. As reported by [111], major operating systems including Android, iOS, Linux, macOS, and Windows have included DoH features in their latest versions. Similarly, popular Internet browsers such as Firefox, Chrome, and Opera also offer DNS encryption options to their users – "Off", "Opportunistic" (or "Automatic"), and "Strict". By the "Off" mode, DNS lookups will be performed via plaintext. For the "Opportunistic" mode, a client and its resolver try to agree on a DNS encryption protocol that is available on both sides. However, plaintext data exchange is used if no agreement can be made. For the "Strict" mode, DNS lookups can only be performed via a certain encryption protocol specified by the user. Their short descriptions are also summarized in Table 2.

However, according to an online survey studying the attitude of clients towards DNS encryption [85], users are quite reluctant and reactive to accept DNS encryption. Majority (67%) of the surveyed population selected the Opportunistic option followed by Strict (20%) and Off (13%) options. This indicates that the adoption of DNS encryption would be highly contingent on how easily application developers enable DNS encryption features on their application, without additional manual configurations from users. One such early example is that Firefox started to bring DoH as its default DNS protocol for their U.S. based users since 2020 [33].

2.2.2 Performance Analysis. Compared with UDP-based DNS in plaintext, encrypted DNS requires additional communication steps for exchanging cryptographic keys (*e.g.,* TLS handshake) and TCP communications (*e.g.,* TCP handshake and payload acknowledgment). Therefore, introducing encryption to DNS inevitably brings more computational and communication overheads that might not be favorable to sensitive applications (*e.g.,* website fetching) [94, 113]. A number of research articles have attempted to quantify the performance implications of DNS encryption by empirically measuring various metrics such as page load time, DNS resolution time, or query response time across a range of access network types (*e.g.,* cellular, campus, or home networks), and resolvers. Their setup details are summarized in Table 3 and will be explained next.

A. Hounsel *et al.* [47] studied the impact of DNS options (plaintext DNS, DoH, and DoT) on the load time of web pages in various networks, including cellular 3G, lossy cellular 4G, or campus wired networks. Their results show that page load times of the three protocols under the (near ideal) university network are almost identical, as their statistical differences approach 0 seconds. Under the 3G network, plaintext DNS performs the best, followed by DoT and DoH. However, DoT gives the best performance in page load time, followed by DoH and plaintext DNS in the lossy

Table 3. A summary of measurement setup in prior works on performance analysis of DNS encryption.

Work	Protocols	Key metrics	Network conditions	Target resolvers
[47]	DNS, DoH, DoT	page loading time	3G, 4G, campus networks	Cloudflare
[22]	DNS, DoH, DoT	query resolution time, page loading time	not specified	local resolver, Google, Cloudflare
[21]	DNS, DoH	page loading time	4G, campus networks	Google, Cloudflare, Quad9
[48]	DNS, DoH, DoT	query resolution time, page loading time	five different ISP networks	Google, Cloudflare, Quad9
[74]	DNS, DoH, DoT	query resolution time, page loading time	mobile, community, and educational networks	local and five public resolvers
[37]	DNS, DoT	query resolution time, failure rate	home networks (mainly in NA & EU)	one local and 15 public resolvers
[49]	DNS, DoH, DoT	query resolution time, latency	home networks in MBA program	three anonymized public resolvers
[29]	DNS, DoH	query resolution time, number of requests per connection	clients from 2190 different autonomous systems	Cloudflare, Google, Quad9, NextDNS

4G networks. The authors believe that possibly it is because TCP (employed by DoT and DoH) has shorter timeout thresholds than UDP (employed by plaintext DNS).

T. Bottger *et al.* [22] quantified the overhead of DNS encryption in resolution time and page load time introduced by the TLS layer (in DoT and DoH) and HTTP layer of versions 1.1 and 2.0 (in DoH). They concluded that both DoT and DoH via HTTP1.0 have significant performance degradation (*e.g.,* additional delays from more than 0.1 to 1 second) due to the head-of-line blocking problem, while DoH using HTTP2.0 is more promising as it results in similar delays compared with plaintext DNS.

K. Borgolte *et al.* [21] investigated the page load time of DoH and plaintext DNS under 4G and campus network scenarios using three different open resolvers by Google, Cloudflare, and Quad9. The authors revealed that only the DoH service of Cloudflare under the university network gives the most reasonable load time. At the same time, other combinations incur significant delays of up to several seconds (*e.g.,* about 5 seconds for DoH using Quad9 resolver under the 4G network). The authors also identified that lossy network conditions and sub-optimal provider selections might widen the performance gaps between DNS and DoH.

A. Hounsel *et al.* [48] highlighted that although plaintext DNS has better performance in query resolution time (*i.e.,* about 300ms and 450ms faster than those of DoT and DoH, respectively), page load times of DoT and DoH are better than that of plaintext DNS (*i.e.,* 101ms and 33ms faster, respectively) in the lossy 4G network. Possibly, this is because it takes a longer time for UDP (plaintext DNS) to detect a lost query, while TCP can quickly react and transmit the lost packet over an existing connection. Given its connection-less property, UDP achieves a shorter response time than TCP in ideal situations (lossless network). In addition, TCP and HTTPS connections initiated by DoT and DoH lookups could be reused by the following web browsing activities to reduce the time overhead introduced by TCP and TLS handshakes. Also, suppose the network becomes heavily lossy (*e.g.,* a 3G network in a remote location). In that case, UDP-based DNS outperforms TCP-based versions (DoH and DoT) since the time for connection establishment would dominate, and UDP would have time to react to losses.

From the above studies, it is quite clear that DNS encryption protocols perform not so well when network conditions are non-ideal. A similar insight was drawn from measurement studies on edge networks [29, 37, 49, 74] (*e.g.,* home) that are far from the essential Internet infrastructures. By measuring DoH resolution times from 22K unique hosts worldwide, R. Chhabral *et al.* [29] found that hosts in high-income countries/regions with better Internet infrastructure are less likely to have performance degradation. At the same time, clients from less-developed areas experience a significant slowdown by switching from plaintext DNS to DoH.

E. S. Mbewe *et al.* measured the performance of DoH, DoT, and plaintext DNS of hosts in Africa [74] and observed high latency and circuitous DNS resolution paths for encrypted DNS packets. For example, round-trip times from South Africa are around 150ms, while the values for Madagascar and Uganda are more than 1000ms.

T. N. Doan *et al.* [37] measured DoT lookups from 3.2K RIPE Atlas probes deployed in home networks. The author first pointed out that only 0.4% of the studied households have their local resolvers supporting DoT. Through extensive measurements, they found higher failure rates (up to 32%) and response times (*e.g.,* about 130ms to 230ms) in DoT than in plaintext DNS, where they observe much lower failure rates (less than 3%) and faster response (less than 100ms).

A. Hounsel *et al.* [49] performed measurements on more than 2500 home networks participating in a Measuring Broadband America program funded by the Federal Communications Commission (FCC). They highlighted that DNS clients could periodically conduct active probing in order to select their optimal user settings in terms of the choice of protocol (DoT or DoH) and resolvers that give a satisfying performance for each home network.

2.3 Benefits of DNS Encryption

Regardless of some performance issues, the industry has advocated encrypted DNS due to its unprecedented benefits such as protecting user privacy and preventing certain attacks that exploit the connectionless property of UDP.

2.3.1 Protecting User Privacy. The risk of end-user privacy leakage is present in various nodes across the path of DNS lookups, shown in Fig. 1. As highlighted by H. Shulman *et al.* [94] and Z. Yan *et al.* [113], eavesdroppers may monitor:

- DNS queries and responses through the links between clients and recursive resolvers.
- DNS logs from recursive resolvers.
- DNS queries and responses through the links between recursive resolvers and authoritative name servers.
- DNS logs from authoritative name servers.

Among these four risks, the first one is of the highest importance since both the client identity and the domain name are exposed to malicious actors who perform passive traffic sniffing. In addition, retrieving logs from recursive resolvers (the second risk listed above) would result in the same outcomes for adversaries; however, it requires them to compromise the servers. On the other hand, plaintext DNS lookups between resolvers and name servers can only reveal an aggregate profile of (*e.g.,* millions of) users served by the public recursive resolver without exposing the identity of individual users. Also, the logs from authoritative name servers do not give away individual user information. DNS encryption protocols, therefore, are primarily designed to protect queries and responses from third-parties, eavesdropping between clients and recursive resolvers [24], as decrypting ciphertext would be largely impractical without knowledge of the secret key.

2.3.2 Preventing Network Attacks on UDP-based DNS. In addition to preserving user privacy, applying encryption techniques to DNS traffic requires the establishment of a secure connection (*e.g.,* TLS, HTTPS, or QUIC) between the client and intended resolver. It changes the connectionless nature of legacy (plaintext) DNS exchanged via port

Table 4. Practical issues and security vulnerabilities of encrypted DNS highlighted by existing literature along with impacted protocols as discussed in §2.4.

Practical issues and security vulnerabilities	References	Protocols
Privacy leakage by compromised resolvers (§2.4.1)	[63, 80, 81, 89]	DoT, DoH, DoQ
Invalid SSL certificates (§2.4.2)	[59, 69]	DoT, DoH
Fallback attacks (§2.4.3)	[54, 93]	DoT, DoH
Imperfect padding strategies (§2.4.4)	[25, 51, 51, 56, 73]	DoT, DoH
Problems of monopolized DNS resolution (§2.4.5)	[50, 65, 88, 91]	DoH
Existing barriers for public adoption (§2.4.6)	[6, 9, 10, 20, 30, 40, 57, 70–72, 97, 113]	DoT, DoH, DoQ

`UDP/53` that may be exploited by malicious actors for various types of network attacks (discussed next). Therefore, applying encryption prevents certain network attacks in the DNS ecosystem. For example, ***DNS amplification attack*** is a popular form of distributed denial of service (DDoS) that relies on the use of publicly accessible open DNS servers to overwhelm a victim system with DNS response traffic [11]. Attackers often craft small-sized DNS queries with source IP addresses spoofed to be a victim's address and send them to resolvers. When resolvers send the DNS record response, it is sent instead to the victim (never requested anything). Attackers will typically submit a request for as much zone information as possible to maximize the amplification effect. Because the size of the response is larger than the request, the attacker is able to increase the amount of traffic directed at the victim. Such attacks are hard to be prevented by the resolvers if DNS queries arrive over connectionless protocol – resolvers are unable to determine whether a query is with spoofed source IP addresses or not. DNS encryption protocols, instead, prevent this problem by requiring a connection to establish via handshaking processes [117]. Therefore, no large-sized DNS response could be sent to a (spoofed) victim IP address.

In addition, L. Zhu *et al.* [117] demonstrated that **direct DDoS attack on DNS servers** is weakened by DNS encryption. This is because attackers need to establish TCP and TLS connections for each attempt of attack (*i.e.,* malicious DNS lookup), leading to higher computing resources required to overwhelm a DNS server. Lastly, encrypting DNS communications between clients and resolvers also reduces the chance of **man-in-the-middle (MITM) DNS spoofing attacks** [82] that aim to mislead (*i.e.,* redirect) clients towards malicious destination IP addresses by hijacking and manipulating DNS responses. As a use-case, P. Jeitner *et al.* [61] developed a secured addressing mechanism using DoH for network time protocol (NTP) systems to prevent the off-path attacks [62] which redirect clients to malicious timing servers via manipulated DNS responses.

2.4 Practical Issues and Security Vulnerability

As highlighted by recent research, although DNS encryption has brought significant enhancements in security and privacy, there are still unsolved issues and security vulnerabilities of encrypted DNS. Some of those include privacy leakage by compromised resolvers, invalid SSL certificates, fallback attacks, imperfect padding strategies, problems of monopolized DNS resolution, and existing barriers for public adoption which are discussed as follows. Table 4 summarizes these issues and vulnerabilities along with their respective prior research and impacted protocols. Note some protocols may truly be subject to a certain category but not listed since they have not been highlighted (reported) by the existing literature. This particularly applies to DoQ, which is relatively a recent encryption protocol compared to other counterparts.

2.4.1 Privacy leakage by Compromised Resolvers. As discussed in §2.3, protocols like DoT, DoH, and DoQ focus on encrypting DNS communications between clients and a recursive resolver, primarily aiming to protect user privacy. Unlike authoritative name servers, recursive resolvers would have the ability to construct a comprehensive profile of users who primarily aimed to keep it private by way of DNS encryption. One may argue that resolver compromise exists with plaintext DNS and continues to exist with encrypted DNS. We note protecting user privacy was not an objective in plaintext DNS, but it is for encrypted DNS protocols. That's why the security of resolvers is of particular interest in the context of encrypted DNS since malicious actors may obtain data of user activities by compromising a recursive resolver. According to L. Jin *et al.* [63], payload manipulation by compromised resolvers is not uncommon even for encrypted DNS. The authors performed more than seven million lookups towards thousands of DoT/DoH-enabled DNS resolvers. They observed that more than 1.5% of responses are manipulated according to their ground-truth records of domain names and the respective IP addresses.

To tackle this potential threat, S. Rivera *et al.* [89] developed a privacy-preserving mechanism that leverages extended Berkeley Packet Filter (eBPF) to assist users in distributing their queries towards a set of resolvers randomly. That way, a compromised resolver would not expose the entire history of query records from a client. Y. Nakatsuka *et al.* [80, 81] developed PDoT, an architecture that hosts recursive resolvers in a Trusted Execution Environment (TEE), so that they can be verified, authenticated, and trusted by end-users before performing DNS lookups.

2.4.2 Invalid SSL Certificates. The SSL certificates of DoT and DoH resolvers are provided to users as proof of their identities prior to the start of encrypted sessions (the process is known as SSL handshake). As reported by C. Liu *et al.* [69], a non-negligible fraction (25%) of 150 DoT servers they studied have invalid SSL certificates that may impose privacy risks to clients as they cannot be verified as trusted entities. Similarly, A. S. Jahromi *et al.* [59] analyzed about $10K$ SSL certificates collected from public DoT servers and found out that only 65% of them are checked as valid. In contrast, the popular reasons for invalidity include non-existent issuers, self-assigned certificates, expired certificates, and expired windows. Note that some of the resolvers with invalid SSL certificates may not be intended for public usage – they may be abandoned servers left from short-term projects on a university network. In summary, given observations from real-world measurements, it is important for service operators to maintain valid certificates when offering encrypted DNS services to the public.

2.4.3 Fallback Attacks. According to §2.2, clients often use the "opportunistic" (or "automatic") mode as their default option to establish encrypted connections with resolvers via the available DNS encryption protocol. Plaintext-based DNS will be used if both sides agree on no encryption protocol. Unfortunately, malicious actors may utilize this fact to steal user privacy by forcing the clients to perform their DNS lookups in plaintext – it is known as a fallback (or rollback/downgrade) attack [93]. Fallback attacks exploit a feature offered by servers operating updated/secured versions of protocols (*e.g.,* DoH and DoT) that may allow a client to communicate via older version counterparts (*e.g.,* plaintext DNS) for backward compatibility. During attacks, a malicious actor often creates fake negotiations (*e.g.,* through man-in-the-middle) between the client and server so that a less-secure protocol is used in the following communications.

As for the viability of fallback attacks, Q. Huang *et al.* [54] reported that four techniques could achieve such fallback, including DNS traffic interception, DNS cache poisoning, TCP traffic interception, or TCP reset injection. The authors have experimented with the four methods on six major browsers such as Chrome and Firefox. Although the response patterns exhibit variations, all tested browsers are found vulnerable to at least one of the four fallback techniques.

Therefore, the authors suggested application developers revisit their encryption policies (*e.g.,* notifying users if the plaintext is chosen) instead of simply leaving “opportunistic” as the default option.

2.4.4 Imperfect Padding Strategies. DNS encryption seems to be promising in protecting privacy by way of ciphertext. However, a recent research [51] found that eavesdroppers can still infer the visited websites of clients with more than 99.5% accuracy through statistical features of DoT traffic such as the temporal patterns of packet sizes.

To prevent data leak from encrypted DNS messages via size-based traffic analysis, the Internet Society proposed a padding strategy in 2018 [73] that fills the DNS queries and responses to a certain size as configured by the users. However, according to K. Hynek *et al.* [56], DNS message padding was not well supported (at least until 2020) by the majority of browsers such as Firefox, thus, its public adoption is inevitably limited.

Regardless of its public adoption, padding encrypted DNS packets is not a perfect solution against data leakage. R. Houser *et al.* [51] managed to achieve more than 80% accuracy in inferring visited websites by analyzing the temporal patterns of padded DoT messages. J. Bushart *et al.* demonstrated in [25] how their method classifies (more than 85%) client device types and predicts (more than 65%) visited domains by combining the size and inter-arrival timing information of padded DoT and DoH from end-users.

2.4.5 Problems of Monopolized DNS Resolution. In the current ecosystem of encrypted DNS, a few major providers such as Google and Cloudflare dominate the market of public resolvers on the Internet. Instead, in plaintext DNS systems, recursive resolvers are well distributed across Internet service providers (ISP) and large public organizations. As discussed next, the related articles pointed out that such a monopolization (also known as centralization) can be detrimental to the reliability and usability of encrypted DNS resolution.

First, having a limited number of major providers dominating the ecosystem can lead to data monopolization [88], which can threaten users' privacy as they would be able to profile user online preference and behavior by their queried domain names. Apart from having more players in this ecosystem, researchers have developed novel methods like Oblivious DNS (ODNS) [91] to alleviate this problem. P. Schmitt *et al.* [91] introduced a client proxy to operate between clients and public resolvers – client proxies receive queries from end-hosts and send them to respective recursive resolvers without revealing their identities. Therefore, with ODNS, monopolized service providers would not have the IP address of their clients. In 2021, ODNS over HTTPS (ODoH) has been standardized as an RFC [65] to prevent the potential misuse of user data by resolver providers. We believe that it is an effective method yet requires public awareness before it is widely adopted.

Second, the monopolized DNS resolution via encrypted protocols provides a sub-optimal quality of service (*e.g.,* resolution time) for user devices that require flexibility in setting their strategy for dynamic selection (depending on geolocation and network conditions) of recursive resolvers. To this end, A. Hounsel *et al.* [50] developed a refactored DNS resolver architecture that supports de-monopolized name resolution, enabling users to specify their lookup preference such as distributing queries across a set of DoH resolvers with certain latency requirements – provides an optimal and flexible selection of resolvers for edge devices. Their prototype evaluation demonstrated an improved performance while preserving user privacy.

2.4.6 Existing Barriers for Public Adoption. The encryption of DNS queries and responses significantly impacts the usability of legacy security measures that inspect plaintext DNS payload for inference and classification purposes. According to the current literature, we now enumerate some security functionalities that have been affected by DNS encryption.

- **Enterprise security via DNS monitoring:** Legacy middleboxes for network security (*e.g.,* border firewalls and network intrusion detection systems) can be configured to inspect DNS messages, gaining visibility into the role of connected assets [71] and/or detecting anomalous/malicious patterns [9, 70], indicative of DNS-based volumetric attacks, accessing illegal contents, or performing data exfiltration. However, payload inspection becomes relatively ineffective with the increasing adoption of encrypted DNS. Hence, new challenges [6] such as lack of complete visibility into DNS traffic, losing control over DNS data, potential leakage of information, and inability to block illegal (inappropriate) contents emerge for enterprise network operators.
- **Internet censorship, parental control, and advertisement blocking:** Internet censorship has been enforced at various levels by the government of many countries such as the U.S., China, and the U.K. [20] to restrict access to certain online services such as pornography, violence, and scandals. Also, monitoring DNS traffic has been used for home networks for parental control purposes [40, 113]. As criticized by relevant stakeholders such as the Internet Watch Foundation [72], with encrypted DNS such as DoH, users can easily bypass censorship imposed at home, organizational, ISP, or national level [30]. In addition, the efficacy of ad-blocking mechanisms, which rely on blocklists of DNS names, is impacted [57].
- **DNS-based criminal investigations:** Internet service providers [113] are often required by laws to record online activities (queried domains) of their subscribers for a certain period that may be needed for criminal investigations. Therefore, serious concerns are raised by agencies like Government Communications Headquarters (GCHQ) that encrypted DNS (as it bypasses current surveillance systems) would cause significant challenges for lawful investigation [30].
- **Captive portal via DNS hijacking:** Commercial venues (*e.g.,* shopping malls and airports) that offer public wireless Internet access to customers often use captive portals to manage their network access through their network gateways. Via DNS hijacking, unauthorized users will be redirected to the login page configured by operators regardless of their actual destinations specified in DNS lookups. The adoption of DNS encryption may render the captive portals ineffective [113] as the network gateways are no longer able to modify DNS responses sent to the end users.
- **Detection of malware distribution:** DNS plays a critical role in the activities of malware. Therefore, security experts have developed methods to detect malware distribution by analyzing domain names in plaintext DNS messages [10, 97]. Unfortunately, those methods are incapable of processing encrypted DNS packets, which do not give visibility into the DNS message content (*e.g.,* query names). This topic will be discussed in detail in §3.

3 MALWARE COMMUNICATIONS EMPOWERED BY DNS ENCRYPTION

Various malware families frequently use DNS for command-and-control (C&C) communications and data exfiltration [83, 87, 100]. While the security community has comprehensively understood such misuse via plaintext DNS, the threat landscape introduced by encrypted DNS has not been well studied yet. This section reviews some of the known ways of malware misusing encrypted DNS by malware and highlights emerging challenges for security teams for detecting those malicious activities. We focus on C&C communications (§3.1) and data exfiltration (§3.2).

3.1 Command-and-Control (C&C) Communications

Malicious actors on the Internet spread their malware to infect connected devices by various methods such as phishing emails, scripts embedded in web pages, and manipulating vulnerable firmware. Those malware-infected devices

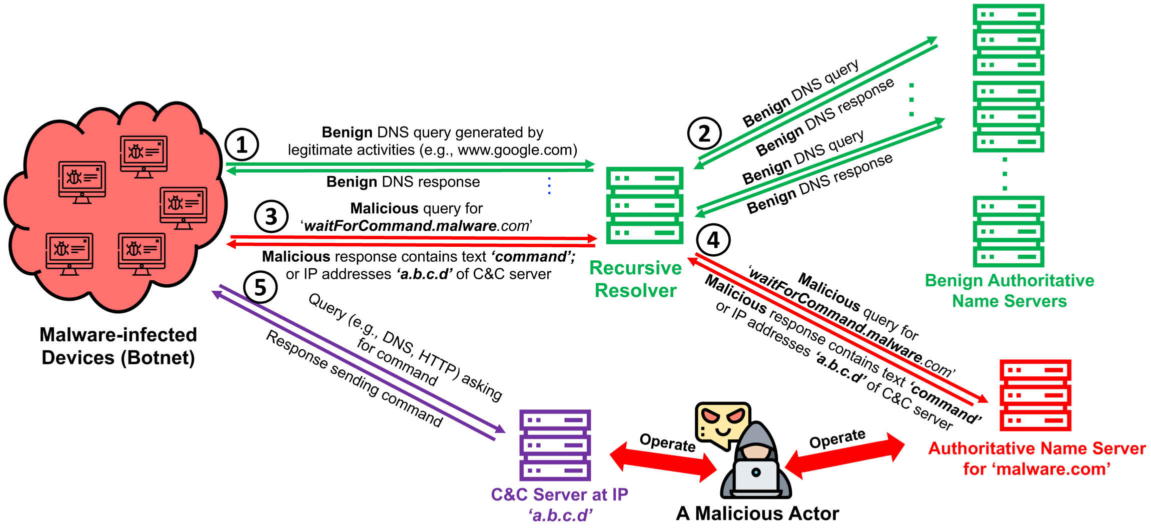
1
Benign DNS query generated by legitimate activities (e.g., www.google.com)
Benign DNS response
3
Malicious query for 'waitForCommand.malware.com'
Malicious response contains text 'command'; or IP addresses 'a.b.c.d' of C&C server
5
Query (e.g., DNS, HTTP) asking for command
Response sending command
Malware-infected Devices (Botnet)
Recursive Resolver
2
Benign DNS query
Benign DNS response
Benign DNS query
Benign DNS response
Benign Authoritative Name Servers
4
Malicious query for 'waitForCommand.malware.com'
Malicious response contains text 'command' or IP addresses 'a.b.c.d' of C&C server
Authoritative Name Server for 'malware.com'
Operate
Operate
A Malicious Actor
C&C Server at IP 'a.b.c.d'

Fig. 4. A visual example of command-and-control (C&C) communications via DNS from malware-infected devices.

form botnets that establish connections with the external malicious actors to conduct cyber crimes as instructed automatically. The process of establishing connections between botnet and masters followed by exchanging information and instructions between these two parties is known as command-and-control (C&C) communications.

3.1.1 C&C Communications via Plaintext DNS. Security appliances and middleboxes (in most enterprise networks) often permit DNS traffic [71]. Attackers, therefore, take advantage of getting free rides and make C&C communications via DNS instead of other proprietary protocols to bypass possible security measures that may block malicious traffic [86].

What is C&C over DNS?: Fig. 4 depicts a typical process of DNS-based C&C communications. In parallel to benign DNS lookups through recursive resolvers (step ① and ②) for legitimate activities, malware-infected devices send DNS queries with intentions embedded in the questioned domain names (*e.g.,* `waitForCommand.malware.com`) towards a malicious authoritative name server (*e.g.,* for `malware.com`) operated by an adversary (step ③ and ④). Note that malicious queries are transmitted to typical DNS recursive resolvers, much like the way for benign DNS lookups. The authoritative name server for `malware.com` may directly send instructions to its botnet via DNS responses with text format (*i.e.,* TXT type). Also, as reported by [112], it is common for other types of C&C communications (*e.g.,* HTTP-based) to locate their masters via DNS resolution, *i.e.,* sending the IP address of a dedicated C&C server (`a.b.c.d` in step ⑤) for the malware-infected device to connect.

What are the countermeasures against C&C over plaintext DNS?: Malware-infected devices that perform regular C&C communications via legacy DNS exhibit relatively distinct patterns that can be used for network monitoring and anomaly detection purposes. The plaintext nature of DNS traffic makes the detection of C&C-related activities reasonably achievable by way of traffic flow analysis and/or packet inspection [68].

For example, features (*e.g.,* temporal activities, composition of domain name strings, DNS header flags) extracted from DNS traffic can be used to detect malware-infected devices. As pointed out by A. Udiyono *et al.* [102], botnet devices (as opposed to benign hosts) often send regular updates via periodic DNS messages on their status to the master servers,

Table 5. Detecting C&C communications: example DNS features and their availability for plaintext and encrypted DNS.

Features	Plaintext DNS	Encrypted DNS
Query packet size [36, 102]	Yes ✓	Yes ✓
Domain TTL value [102]	Yes ✓	No ✗
Number of distinct domain names [19, 102]	Yes ✓	No ✗
Number of client IP addresses [19]	Yes ✓	Yes ✓
Temporal and spatial query pattern [19, 36]	Yes ✓	Yes ✓
Fraction of numerical characters in domain names [19]	Yes ✓	No ✗
Length of the longest meaningful substring (LMS) [19]	Yes ✓	No ✗
Entropy of domain names [36]	Yes ✓	No ✗
Volume of NXDOMAIN response [109]	Yes ✓	No ✗

resulting in a high volume of DNS queries. Also, domain names owned and operated by malicious actors are likely to be short-lived with low TTL values (*i.e.,* also known as disposable domains [115]). That way, they can frequently change their DNS mapping and hence bypass static measures like blocklisting. In addition, devices that are infected by certain types of malware query for a relatively high number of distinct domain names due to the use of domain generation algorithms (DGA) that automatically emits a sophisticated set of domain names as the rendezvous points with C&C servers. Using the DNS features discussed above, A. Udiyono *et al.* [102] were able to detect botnet traffic with a relatively high accuracy (≈90%).

DNS features have proven effective in detecting malicious domain names used in C&C communications. L. Bilge *et al.* [19] extracts 15 DNS traffic features of four categories, including time-based, DNS answer-based, TTL value-based, and domain name-based properties, which collectively detect malicious domain names used for C&C. The authors developed machine learning models (trained on the identified features) with a large dataset containing over 100 billion DNS requests collected from an ISP network. The evaluation results show a close to perfect accuracy (more than 99.5%) can be achieved in detecting malicious domains.

In addition, by employing attributes of DNS traffic, researchers have identified a group of hosts managed certain malware families. C. J. Dietrich *et al.* [36] shows how traffic characteristics like the entropy of queried domain names or aggregate behavioral activity profiles (*e.g.,* DNS response rates) enabled the authors to detect C&C via DNS and determine their malware families.

3.1.2 C&C Communications via Encrypted DNS. Recent evidence from the security community [87] has shown that malware developers are increasingly exploiting encrypted DNS (*i.e.,* DoH) for C&C communications, bypassing the existing tools for security monitoring. For example, in 2019, the security team of 360 Netlab reported the existence of Godlua backdoor malware [83] that uses DoH for C&C communications. According to [83], several vendors mark Godlua as mining-related trojan malware while also involved in DDoS attacks. Malware strains that use encrypted DNS for C&C communications continue to evolve and display more dynamic and stealthy behaviors to reduce their chance of being detected. As reported by the Proofprint Threat Insight Team [100], the ".NET-based" malware PsiXBot that emerged to perform C&C communications via the resolution of certain malicious domains now uses Google's public DoH resolver, and hence does not look suspicious anymore. Therefore, network security appliances become ineffective when applying their pre-populated blocklists to those malicious C&C-related queries.

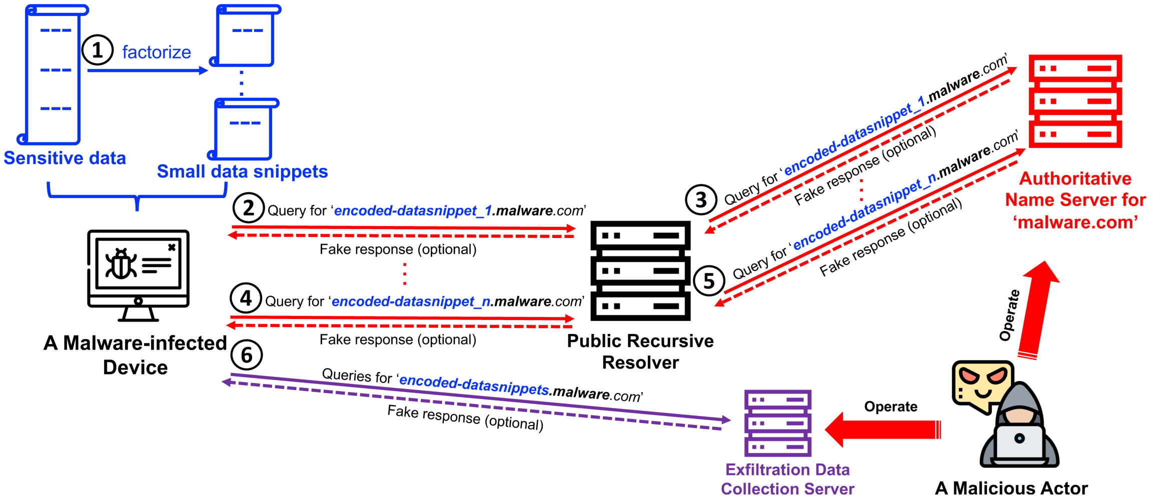
① factorize
Sensitive data
Small data snippets
A Malware-infected Device
② Query for 'encoded-datasnippet_1.malware.com'
Fake response (optional)
④ Query for 'encoded-datasnippet_n.malware.com'
Fake response (optional)
⑥ Queries for 'encoded-datasnippets.malware.com'
Fake response (optional)
Public Recursive Resolver
③ Query for 'encoded-datasnippet_1.malware.com'
Fake response (optional)
⑤ Query for 'encoded-datasnippet_n.malware.com'
Fake response (optional)
Authoritative Name Server for 'malware.com'
Operate
Operate
Exfiltration Data Collection Server
A Malicious Actor

Fig. 5. A visual example of data exfiltration via DNS from malware-infected devices.

Implications of encryption for C&C detection: The trend of C&C via encrypted DNS inevitably introduces significant challenges to defenders as they become unable to extract most of the key DNS features as in legacy plaintext DNS [24]. Table 5 summarizes key features for detecting C&C communications over plaintext DNS traffic (already discussed in §3.1.1) and challenges (lack of their availability) when applied to encrypted DNS.

Unsurprisingly, those features that require inspection of DNS packet headers and payloads (*e.g.,* query name, query type, error code, resource records, and domain TTL) are not available anymore due to content encryption. In contrast, obtaining features from unencrypted IP headers (*e.g.,* client IP addresses in Table 5), aggregate network profiles (*e.g.,* query packet volume), and time-series patterns [76] may still be feasible [43] for detecting C&C over encrypted DNS.

3.2 Data Exfiltration (Tunneling)

Malware may retrieve sensitive and critical data (*e.g.,* user credentials, credit card details, business financial, or medical records) from the infected devices and send it to unauthorized external entities. This process is known as "data exfiltration" by the security community [103].

3.2.1 Data Exfiltration via Plaintext DNS. Commercial security middleware and appliances, operation on the network of ISPs and/or enterprises, may be empowered to prevent or detect data thefts [31] by way of traffic inspection. However, DNS communication is relatively poorly policed by organizations (compared to services like email, FTP, and HTTP) and has been exploited by cyber-criminals to maintain covert communication channels with compromised hosts. Such stealthy methods, leveraging the DNS protocol, is known as DNS exfiltration or tunneling [79] .

DNS exfiltration: Fig. 5 illustrates a typical process of DNS exfiltration, with a sequence of events annotated by circled numbers. For the first step (the top-left region of Fig. 5), a malware-infected device factorizes the sensitive data into small snippets that are suitable to be encoded into (the subdomain of) DNS query names. From step ② to ⑤, the device crafts and sends a series of DNS queries with their subdomain names carrying the encoded data (*i.e.,* `encoded-datasnippet_1` till `encoded-datasnippet_n`) belonging to a malicious domain (*i.e.,* `malware.com`) managed by an external attacker. These queries reach the malicious name server (`malware.com`) through private/public

Table 6. Examples of a short benign domain name, a long benign domain name, and a data exfiltration domain name [9].

Type	**Domain name**
Benign	`www.google.com`
Benign	`p4-ces3lawazdkbw-qlrq5qalxdt7ytcq-385202-i1-v6exp3.ds.metric.gstatic.com`
Exfil.	`PzMnPios0D3n0Cwu0zomPS4nNjovPS8u0zsnNCst0Dkj0cwoMwAA.29a.de`

recursive resolvers and might get dummy responses as the acknowledgments of received data. Exfiltrating data via DNS recursive resolvers makes the delivery path seemingly legitimate, however, recursive resolvers may block some of the reported primary domains to stop such misuses. We note that malicious actors may choose to directly send exfiltration DNS queries to a collection server managed by them (step ⑥ in Fig. 5). Finally, upon arrival of all malicious DNS queries (sourced from the infected device), the attacker can reconstruct the exfiltrated data file by decoding and combining all snippets extracted from the subdomain names.

Countermeasures against data exfiltration via plaintext DNS: Research works on detection of data exfiltration over plaintext DNS have proven to be successful. They generally employed statistical features extracted from query names to train machine learning classifiers [90] to distinguish benign from malicious queries.

Let us have a look at some of these queries. Table 6 shows three sample query names (benign and exfiltration), as reported by by J. Ahmed *et al.* [9], illustrating the difference between benign and malicious query names in the context of plaintext DNS. We note that such distinction may not be evident with encrypted DNS queries. It can be seen the exfiltrated query name (the thrid row highlighted in red) contains a fairly long subdomain string with 52 English letters, whereas a typical benign subdomain only contains several English letters (*e.g.,* three for `www` in the first row). Note that some benign domain names may also have a long subdomain string (the second row highlighted in blue). However, exfiltration strings generated by encoding algorithms look random with a mix of numbers and uppercase/lowercase letters. In contrast, the long benign subdomains often display more meaningful patterns indicative of their service categories/indexes (the first row highlighted in green).

Based on large-scale data analysis of real queries for both benign and suspicious/malicious domains, J. Ahmed *et al.* [9] identified eight features of individual domain names to detect anomalous string patterns in exfiltration DNS queries. They used features like the count of uppercase characters and characters in subdomain names. The authors trained a one-class classifier on benign data that achieved an overall detection accuracy of more than 98%.

Using a different approach, work in [90] focused on other aspects of the DNS traffic, considering features like spatial and temporal statistics of query packets from a given host to detecting DNS exfiltration from malware-infected hosts.

3.2.2 Data Exfiltration via Encrypted DNS.

The deployment of DNS encryption techniques and their supporting infrastructures (*e.g.,* encryption-enabled public resolvers) provide a solid basis for data exfiltration over encrypted DNS protocols. Therefore, it is not surprising to witness their occurrence, such as the one described in [28]. Oilrig (or APT34), a hacking group, was reported in 2020 as the first known malicious actor that managed to perform data exfiltration over DoH by advancing an open-source exfiltration toolkit [1]. The victim of that DoH-based data exfiltration may include a pharmacy company when it announced its commencement of research for the treatment of COVID-19, according to [28]. Many open-source tools [4] exist for data exfiltration over DoH on the Internet and hence becomes relatively easy for malicious actors. Therefore, enterprises that host critical data are alarmed [31] to stay vigilant for emerging threats before getting harmed.

Table 7. Detecting data exfiltration: representative DNS features extracted from domain names and their availability in plaintext and encrypted DNS traffic.

Features	**Plaintext DNS**	**Encrypted DNS**
Total count of characters in fully qualified domain names (FQDN) [9, 44]	Yes ✓	No ✗
Count of characters in sub-domain names [9]	Yes ✓	No ✗
Count of uppercase characters in domain names [9]	Yes ✓	No ✗
Count of numerical characters in domain names [9]	Yes ✓	No ✗
Entropy of domain names [9, 44]	Yes ✓	No ✗
Number of domain name labels [9]	Yes ✓	No ✗
Maximum domain name labels [9]	Yes ✓	No ✗
Average domain name labels [9]	Yes ✓	No ✗

How DNS encryption impacts data exfiltration detection: The use of encrypted DNS in data exfiltration attacks inevitably makes detecting them more difficult than those leveraging plaintext DNS. Methods for detecting C&C over encrypted DNS may still be able to resort to patterns in signaling packets (*i.e.,* malware-infected devices are likely to connect with their C&C servers periodically [52]). However, existing methods for detecting DNS data exfiltration that primarily analyze the string patterns in query names become ineffective after encryption. Table 7 summarizes features[1] extracted from query names [9, 44] that can recognize (with decent confidence) the DNS exfiltration over plaintext DNS. It can be seen in the last column of this table that all required features become unavailable after DNS encryption.

4 INFERENCE TECHNIQUES FOR ENCRYPTED DNS TRAFFIC

As discussed in §3, malware leverages encrypted DNS protocol to bypass inspection-based detection and mitigation techniques used for C&C communications and data exfiltration. Although the current countermeasures against such misuses have not matured, some promising attempts to analyze encrypted DNS traffic for classification (not necessary for security purposes) provide useful lessons and references. In this section, we start by elaborating on the current research works in detecting encrypted DNS packets from HTTPS traffic streams and classifying their types (*e.g.,* benign or malicious) in §4.1. Next, we review the developed methods in fingerprinting user profiles by analyzing encrypted DNS traffic, which can be a useful reference in developing methods to identify malware-infected hosts (§4.2).

4.1 Detecting and Classifying Encrypted DNS Traffic

Plaintext DNS, DoT, and DoQ have their dedicated service port numbers as `UDP/53`, `TCP/853`, and `UDP/784&8853`, respectively, which can be easily detected by inspecting headers. However, DoH encapsulates its DNS content in HTTPS packets, and thus it shares the same service port `TCP/443` with other HTTPS-based applications (*e.g.,* web browsing). Unsurprisingly, the detection of DoH packets from HTTPS traffic is a nontrivial problem.

One may label an HTTPS flow as DoH if its source or destination IP address is one of the well-known DoH recursive resolvers. However, such methods inevitably: (a) mislabel non-DoH HTTPS flows towards the IP address of popular resolvers (*e.g.,* Google's DoH resolver shares the IP address `8.8.8.8` with an HTTPS-based website [42] for manual DNS lookups via a web interface), and (b) will miss DoH flows to unpopular servers. As pointed out in [87], the analysis

[1] The entropy of domain names has already been discussed in §3.1 for C&C detection.

Table 8. Key flow features (summarized from information in [108]) and their expected value range for DNS over HTTPS (DoH), web browsing, file downloading, and video streaming.

Features	**DNS over HTTPS**	**Web browsing**	**File downloading**	**Video streaming**
Flow duration	Long	Short	Long	Long
Flow volume	Small	Medium	Large	Large
Variance of packet sizes	Small	Large	Medium	Large
Ratio of burst (active) and pause (idle)	High	Small	Small	High
Number of packets in each burst	Medium	High	High	High
Symmetry of requests and responses	High	Low	Low	Low

of traffic patterns seems to be a promising approach to detect and classify DoH packets due to some of their distinct characteristics. In what follows, we discuss the related research efforts in more detail.

4.1.1 Detecting DoH from HTTPS Traffic. D. Vekshin *et al.* [108] pointed out that accurate recognition of DoH is a precursor step for detecting encrypted DNS-based malware activities. They explored the possibility of using machine learning techniques to differentiate DoH from HTTPS traffic. To this end, the authors analyzed the identifiable flow characteristics of DoH. First, clients often establish a single DoH connection to their recursive resolver and retain it for a number of domain lookups. Therefore, the flow duration of DoH is likely to be longer than that of typical HTTPS connections (*e.g.*, web browsing). Although video streaming and large file downloading via HTTPS may also have long durations, their volume and data rate are expected to be higher than DoH flows. Second, the variance of DoH packet sizes is much lower than other HTTPS applications. Third, DoH packets often come in bursts during a connection depending on users' interactions, but the total number of DNS packets (and the corresponding volume) in each burst is not large. Finally, unlike other applications, a DoH flow often has quite similar (or symmetric) volume in both directions.

Observing some of unique characteristics of DoH flows (mentioned above) as well as their temporal behaviors, D. Vekshin *et al.* [108] identified 19 features capturing identifiable patterns in DoH flows. We summarize and aggregate in Table 8 those key features and show their expected value range for DoH and three typical HTTPS-based applications, including web browsing, file downloading, and video conferencing. With those key flow-level attributes, the authors developed machine learning classifiers to distinguish DoH from generic HTTPS traffic that yields a very high accuracy of more than 99% on their dataset publicly available at [107]. Similarly, L. Csikor *et al.* [32] built their machine learning models using features of packet sizes and inter-arrival times that can differentiate DoH from website browsing with more than 90% accuracy.

4.1.2 Classifying Benign and Malicious DoH Traffic. With the rising use of DoH in malicious activities such as C&C and data exfiltration & tunneling, classifying benign and malicious DoH traffic becomes increasingly essential.

C. Patsakis *et al.* [87] performed time-series modeling via Hodrick-Prescott (HP) filter on DoT and DoH response sizes from benign and malware-infected hosts. They observed distinct patterns in time-series signals of packet size to identify hosts performing C&C communications and classify their malware families.

C. Kwan *et al.* [67] developed a threshold-based method on packet size, packet rate, and throughput to differentiate DoH tunneling from benign DoH communications. Their method achieved a perfect 100% accuracy in detecting DoH tunneling generated by a popular tool called "dnstt" [3]. The authors demonstrated that DNS tunneling could only bypass their detection method if its rate is significantly reduced by a factor of 27.

M. MontazeriShatoori *et al.* [76] detects DoH tunneling by combining time-series (*i.e.,* a sequence of packet size, count, duration, and inter-arrival time), header (*i.e.,* unencrypted handshake information), and statistical features (*e.g.,* mean, median, and variance of packet sizes) of HTTPS traffic. The authors make their dataset "CIRA-CIC-DoHBrw-2020" publicly available [26], containing more than 100M packets of generic HTTPS, benign DoH, and malicious DoH. Their traffic traces are generated by two browsers (Chrome and Firefox), three DoH exfiltration tools (Iodine, DNS2TCP, and DNScat2), and four public DoH resolvers (Adguard, Cloudflare, Google, and Quad9). The authors developed machine-learning models with a hybrid set of features from a rich and diversified dataset that achieved over 99% precision in malicious DoH traffic detection. Using the "CIRA-CIC-DoHBrw-2020" dataset, three follow-up research works [17, 18, 98] demonstrated the effectiveness of various machine learning algorithms such as Logistic Regression (LR), Random Forest (RF), K-Nearest Neighbors (KNN), XGBoost, Light gradient boosting machine (LGBM) using 34 traffic features originally identified by M. MontazeriShatoori *et al.* [76].

4.2 Profiling User Activity by Analyzing Encrypted DNS Traffic

In addition to the detection of encrypted DNS traffic from generic HTTPS traffic, some emerging research works analyze encrypted DNS traffic to profile the behavior of hosts (*i.e.,* users) – insights such as websites they visit and operating systems they use. To the best of our knowledge, there is no such work for detecting malware-infected hosts. That said, current research arts provide valuable references and insights for solving cybersecurity problems. Therefore, we discuss them as follows.

R. Houser *et al.* in [51] fingerprints the history of visited websites by hosts using temporal patterns of DoT packet sequences. The authors identify important features that fall into nine categories, including length of query and response, number of queries and responses, volume of queries and responses, time intervals, total transmission time, sequence of DNS packets of uninterrupted queries and responses, number of DNS messages in each TLS record, query rate, and time to receive the first N bytes from the resolver. By selecting and tuning optimal models, the authors are able to achieve decent performance, false negative and false positive rates of less than 17% and 0.5%, respectively, in identifying the visited website of a host. Moreover, they showed how their techniques work for padded DoT messages [73] with around 99% true positive rate and 42% true negative rates.

Similarly, S. Siby *et al.* [95, 96] proposed a machine learning-based method to fingerprint website visits of a host by analyzing DoH traffic. For a DoH flow, they use n-grams (*i.e.,* a contiguous sequence of n queries and responses) of TLS record lengths and burst-lengths (*i.e.,* the total length of consecutive packets in the same direction) as key features to train a random forest classifier. Evaluation results showed that their approach achieved more than 90% accuracy in fingerprinting website visits. The authors demonstrated that client locations, different DoH resolvers, client operating systems, and browser applications only have a minor impact on the classification performance. However, the use of perfect padding (currently not widely adopted by the stakeholders) deteriorates their model's prediction, giving a poor accuracy of less than 10%.

Apart from website fingerprinting, Segram, as presented by M. Muhlhauser *et al.* [78], is able to identify Android applications for smartphones and IoT devices used by clients through the analysis of their DoT or DoH traffic. Apart from website fingerprinting, Segram [78] is able to identify Android applications for smartphones and IoT devices used by clients through the analysis of their DoT or DoH traffic. The authors used the n-grams of DNS sequences (message sizes and interarrival time) as primary features of their classification model, which outperforms their counterparts [96]

Table 9. Key features and their applicability & effectiveness (reported in [51, 78, 96]) for user profiling when extracted from padded/unpadded DoT/DoH traffic.

Features	Unpadded DoH	Unpadded DoT	Padded DoH	Padded DoT
Query and response length [51]	Applicable	Applicable	Not effective	Not effective
Number of queries and responses [51]	Applicable	Applicable	Applicable	Applicable
Inter-arrival time [51]	Applicable	Applicable	Applicable	Applicable
Transmission time [51]	Applicable	Applicable	Applicable	Applicable
Order of query and response packets [51]	Applicable	Applicable	Applicable	Applicable
Volume of queries and responses [51]	Applicable	Applicable	Not effective	Not effective
Number of DNS message in TLS records [51]	Not applicable	Applicable	Not applicable	Applicable
Query rate [51]	Applicable	Applicable	Applicable	Applicable
Time to receive the first *N* bytes [51]	Not applicable	Applicable	Not applicable	Applicable
N-grams of TLS record lengths [96]	Applicable	Not effective	Not effective	Not effective
N-grams of burst sizes [96]	Applicable	Not effective	Not effective	Not effective
N-grams of DNS sequences [78]	Applicable	Applicable	Not effective	Not effective

employing other features such as n-grams of TLS record sizes and burst lengths. More specifically, their developed model achieved more than 90% accuracy on the traffic traces (publicly available at [77]) of 118 Android apps from ten DoT/DoH resolvers. They highlighted that their method is proven to be relatively effective when applied to padded encrypted DNS, giving over 72% classification accuracy.

In addition to profiling the user behaviors by analyzing encrypted DNS traffic on the network, G. Varshney *et al.* [106] showed that how they can obtain DoH lookups before they get encrypted by passively monitoring the RAM usage on the client devices. It is important to note that DNS encryption promises to protect data privacy in transit, but organizations and/or users cannot expect it to prevent such strong attacks when a user device is infected/compromised.

In Table 9, we summarize the applicability and efficacy of key features of unpadded DoH, unpadded DoT, padded DoH, and padded DoT traffic in order to profile users online activity as reported by the above research works. For a certain traffic type, a feature is labeled as "applicable" if it can be extracted; otherwise, it is labeled as "not applicable". In addition, a feature that can be computed but leads to relatively poor classification performance (*i.e.,* accuracy less than 80% as reported by the respective literature) is marked as "not effective" in Table 9. For example, as stated by M. Muhlhauser *et al.* [78], classifications using n-grams of DNS sequences can only achieve 72% accuracy for padded DoH and DoT; thus, they are labeled as "not effective" in the respective cells. We believe that those important characteristics of encrypted DNS traffic are still valuable in future user profiling works for cybersecurity use-cases like detecting malware-infected hosts.

5 DISCUSSION AND FUTURE DIRECTIONS

With the understanding of current development and issues of DNS encryption and some of the risks of malware misuse, we now discuss potential research directions worthwhile to be explored in future works.

5.1 Performance Enhancements

Although DNS encryption promises privacy and security benefits to users and service providers, overheads of TCP acknowledgments and TCP/SSL handshakes in DoT and DoH can led to some performance degradation (*e.g.,* higher query resolution times are observed in the distribution of real-world measurements discussed in §2.2.2, such as by A. Hounsel *et al.* [50]) compared with plaintext DNS over UDP. Such performance degradation is more evident to clients under unstable network conditions in rural areas and/or wireless networks. Thus, it becomes one of the key reasons for the low adoption of DNS encryption. DoQ, which is currently in the draft stage, seems to be a faster DNS encryption protocol due to its zero-RTT feature. Apart from the protocol-level optimization, future research for performance enhancement mechanisms on client applications (*e.g.,* browsers), network equipment (*e.g.,* routers), and resolvers is necessary to boost the public adoption of DNS encryption.

5.2 Managing Security and Privacy Vulnerability

As discussed in §2.4 , there are still privacy issues that remain unresolved or are newly introduced by DNS encryption. We now recap/highlight those issues and outline some potential directions of research to address them.

First, although DNS communications between clients and resolvers are secured against third-party eavesdropping, a compromised (or malicious) resolver could still put user privacy at risk. Furthermore, current DoT/DoH/DoQ resolvers are dominated by a limited number of servers owned and operated by major service providers. Such centralization could lead to data monopolization that ultimately harms clients' privacy. Therefore, developing privacy-preserving methods against data leakage from the resolver is a valuable direction – ODNS [91] that detaches the proxy function from resolvers is an example of such methods.

Second, compared with plaintext DNS resolvers, DoH and DoT resolvers become vulnerable to a broader range of attacks such as TCP and TLS-based DDoS attacks in addition to typical DNS query floods given their need for establishing TLS/TCP connections with clients (legitimate or malicious). In addition, DNS encryption is not a silver bullet to prevent all types of attacks over DNS, such as injection attack [60]. Undoubtedly, effectively securing the key infrastructure (*i.e.,* encrypted DNS resolvers) from potential DDoS attacks requires more sophisticated protection, mostly related to the management of TCP and TLS connections, than plaintext DNS.

Third, misconfiguring encryption settings on either client (*e.g.,* choice of encryption option) or resolver (*e.g.,* invalid SSL certificate or insecure TLS version) make the privacy protection promised by DNS encryption less effective. Therefore, a systematic way of sanity checking and/or verification on DNS encryption settings for user applications (*e.g.,* browsers and mobile applications) and resolver can be an avenue for research to explore.

Last, third-party observers (or eavesdroppers) may still identify user profiles (*e.g.,* visited websites and used applications) by analyzing the traffic patterns of their encrypted DNS communications. Malicious usage of such techniques may result in leaking clients' private information. Some potential countermeasures such as privacy-preserving query scheduling techniques (*e.g.,* [12]) that obfuscate DNS lookup patterns of various applications are worthwhile to be further developed, protecting user privacy.

5.3 Understanding Malicious Encrypted DNS Traffic

As discussed in §3, encrypted DNS has been misused for malware C&C communications and data exfiltration to evade security appliances and middleware that perform payload inspection, looking for malicious signatures. With the increasing growth of DNS encryption, its misuse by malicious actors will likely become more frequent and diversified.

Hence, there is a need for advanced methods to characterize malicious encrypted DNS communications, addressing emerging threats.

5.4 Host Profiling Techniques for Malware Detection

As discussed in §4.2, current host profiling techniques that analyze encrypted DNS traffic mainly focus on identifying visited websites and mobile applications. Existing works have shown that hosts display distinguishable patterns of encrypted DNS traffic when they perform certain activities (*e.g.,* visiting different websites). They extract a comprehensive set of features, achieving acceptable accuracy in their respective tasks. We believe such profiling techniques can be employed to detect malware-infected hosts. For example, quantifying deviations from their baseline benign profile (anomaly detection) or employing multi-class classifications to identify specific types of malware misuses (*e.g.,* classes of malware or types of misuse) could be considered.

We note that the features (summarized in Table 9), proven to be effective for other tasks, may also be useful in detecting malware activities via encrypted DNS.

6 RELATED SURVEYS ON DNS SECURITY

DNS security has been a popular topic, with many published surveys, each focusing on a specific topic ranging from the categorization of DNS attacks to methods for detecting DNS attacks or identifying certain types of DNS attacks. We discuss (in chronological order) some of the contemporary ones published after 2016.

Y. Zhauniarovich *et al.* [116] reviewed data-driven techniques for detecting malicious domains with a focus on aspects of dataset collection (*i.e.,* the sources of DNS data, data enrichment, and ground truth), algorithm design (*i.e.,* feature engineering, detection mechanisms, and outcomes), and evaluation methodologies (*i.e.,* performance metrics and strategies). S. Torabi *et al.* [101] surveyed passive systems of DNS traffic analysis for detecting various DNS attacks across the Internet, including DNS protocol attacks (*e.g.,* DNS spoofing), DNS server attacks (*e.g.,* DDoS), and DNS abuse (*e.g.,* C&C communications). The authors looked at various system designs and compared their objectives, scopes, detection approaches, analyzed traffic features, dataset, and evaluation methods. The survey also highlighted the practicality of the studied systems by considering their real-time performance. N. U. Aijaz *et al.* [104] studied DNS vulnerabilities, including cache poisoning attack, DDoS attack, spoofing, phishing, identity theft, forgery, eavesdropping, packet sniffing, tampering, and man-in-the-middle attack. In their discussions, they also included the protocol-level security enhancements introduced for DNS (*i.e.,* DNSSEC, SSL certificates, and extended validation certificate). Y. Wang *et al.* [109] surveyed (plaintext) DNS tunneling detection methods developed during years between 2006 and 2020. They highlight the statistical features (*i.e.,* payload-based and traffic-based) and detection mechanisms (*i.e.,* rule-based and model-based) used in each DNS tunnel detection method.

Our survey focuses on DNS encryption and highlights opportunities (*i.e.,* protecting privacy) and risks (*e.g.,* malware misuse) associated with encrypted DNS traffic. To the best of our knowledge, there is no survey on the topic of DNS encryption to date. We found all existing relevant survey papers only cover the landscape of plaintext DNS.

7 CONCLUSION

This paper conducted a systematic and comprehensive review of academic research papers and industrial reports on the development and current status of DNS encryption, with a specific focus on its misuse by malware and potential countermeasure techniques. We outlined the development of three DNS encryption protocols (including DoT, DoH, and DoQ), their current state of public adoption, and their performance. We discussed the security benefits and risks of

DNS encryption techniques highlighted by the current literature. Among security risks, we particularly focused on the potential malware misuses of encrypted DNS protocols, including C&C and data exfiltration. Although detecting malicious activities over plaintext DNS has proven relatively successful by prior works using inspection DNS payloads, their applicability to encrypted traffic is yet to be determined. Next, we studied existing works on the analysis of encrypted DNS traffic that can detect malware activities over encrypted DNS. The works are classified as either detecting encryption DNS packets from generic HTTPS traffic and classifying their types (*e.g.,* malicious or benign), or fingerprinting user profiles by analyzing encrypted DNS traffic of hosts. Those works provide solid starting points and valuable references for malware detection potentially applicable to encrypted DNS. Inspired by prior works in the literature, we identified directions for future works, including performance enhancement, managing security and privacy issues, understanding the misuse of encrypted DNS by malware, and developing host profiling techniques to detect malware activities over encrypted DNS protocols.

REFERENCES

[1] 2018. DNSExfiltrator. https://github.com/Arno0x/DNSExfiltrator. Accessed: 2021-11-01.

[2] 2018. HTTPS-Only Features in Major Browsers. https://www.digicert.com/blog/https-only-features-in-browsers. Accessed: 2021-10-12.

[3] 2019. DNSTT. https://github.com/Mygod/dnstt. Accessed: 2021-10-12.

[4] 2019. Research into Data Exfiltration using DOH. https://sysopfb.github.io/exfiltration,/c2/2019/09/22/DOH-exfiltration.html. Accessed: 2021-10-12.

[5] 2021. DNSCrypt & DoH Servers. https://dnscrypt.info/public-servers/. Accessed: 2021-10-19.

[6] 2021. How DNS-over-HTTPS (DoH) has Changed the Threat Landscape For Companies. https://quointelligence.eu/2021/02/dns-over-https-doh/. Accessed: 2021-10-12.

[7] 2021. Public Resolvers. https://dnsprivacy.org/public_resolvers/. Accessed: 2021-10-22.

[8] 2021. Publicly Available Servers. https://github.com/curl/curl/wiki/DNS-over-HTTPS. Accessed: 2021-10-22.

[9] Jawad Ahmed, Hassan Habibi Gharakheili, Qasim Raza, Craig Russell, and Vijay Sivaraman. 2020. Monitoring Enterprise DNS Queries for Detecting Data Exfiltration From Internal Hosts. *IEEE Transactions on Network and Service Management* 17, 1 (Mar 2020), 265–279.

[10] Kamal Alieyan, Ammar Almomani, Ahmad Manasrah, and Mohammed M. Kadhum. 2017. A Survey of Botnet Detection Based on DNS. *Neural Comput. Appl.* 28, 7 (Jul 2017).

[11] Marios Anagnostopoulos, Georgios Kambourakis, Panagiotis Kopanos, Georgios Louloudakis, and Stefanos Gritzalis. 2013. DNS Amplification Attack Revisited. *Computers & Security* 39 (Nov 2013), 475–485.

[12] Oscar Arana, Hector Benítez-Pérez, Javier Gomez, and Miguel Lopez-Guerrero. 2021. Never Query Alone: A distributed strategy to protect Internet users from DNS fingerprinting attacks. *Computer Networks* (Nov 2021).

[13] R. Arends, R. Austein, M. Larson, D. Massey, and S. Rose. 2005. *DNS Security Introduction and Requirements.* RFC 4033. https://datatracker.ietf.org/doc/html/rfc4033

[14] Vasily Bagirov. 2020. AdGuard DNS-over-QUIC. https://adguard.com/en/blog/dns-over-quic.html. Accessed: 2021-10-22.

[15] Xiaolong Bai, Liang Hu, Zixing Song, Feiyan Chen Chen, and Kuo Zhao. 2011. Defense against DNS Man-In-The-Middle Spoofing. In *Proc. WISM.* Taiyuan, China.

[16] Hitesh Ballani and Paul Francis. 2008. Mitigating DNS DoS Attacks. In *Proc. ACM CCS.* Alexandria, Virginia, USA.

[17] Yaser M. Banadaki. 2020. Detecting Malicious DNS over HTTPS Traffic in Domain Name System using Machine Learning Classifiers. *Journal of Computer Sciences and Applications* 8, 2 (Aug 2020), 46–55.

[18] Matthew Behnke, Nathan Briner, Drake Cullen, Katelynn Schwerdtfeger, Jackson Warren, Ram Basnet, and Tenzin Doleck. 2021. Feature Engineering and Machine Learning Model Comparison for Malicious Activity Detection in the DNS-Over-HTTPS Protocol. *IEEE Access* (Sep 2021).

[19] Leyla Bilge, Engin Kirda, Christopher Kruegel, and Marco Balduzzi. 2011. EXPOSURE: Finding Malicious Domains Using Passive DNS Analysis. In *Proc. NDSS.* San Diego, California, USA.

[20] Paul Bischoff. 2021. Internet Censorship 2021: A Global Map of Internet Restrictions. https://www.comparitech.com/blog/vpn-privacy/internet-censorship-map/. Accessed: 2021-10-22.

[21] Kevin Borgolte, Tithi Chattopadhyay, Nick Feamster, Mihir Kshirsagar, Jordan Holland, Austin Hounsel, and Paul Schmitt. 2019. How DNS over HTTPS is Reshaping Privacy, Performance, and Policy in the Internet Ecosystem. *SSRN Electronic Journal* (Jan 2019).

[22] Timm Böttger, Felix Cuadrado, Gianni Antichi, Eder Leão Fernandes, Gareth Tyson, Ignacio Castro, and Steve Uhlig. 2019. An Empirical Study of the Cost of DNS-over-HTTPS. In *Proc. ACM IMC.* Amsterdam, Netherlands.

[23] Theogene Hakiza Bucuti and Ram Dantu. 2015. An Opportunistic Encryption Extension for the DNS Protocol. In *Proc. IEEE ISI.* Baltimore, MD, USA.

[24] Kimo Bumanglag and Houssain Kettani. 2020. On the Impact of DNS Over HTTPS Paradigm on Cyber Systems. In *Proc. ICICT.* San Jose, CA, USA.

[25] Jonas Bushart and Christian Rossow. 2020. Padding Ain't Enough: Assessing the Privacy Guarantees of Encrypted DNS. In *Proc. USENIX FOCI*. Santa Clara, CA, USA.

[26] Canadian Institute for Cybersecurity (CIC) project funded by Canadian Internet Registration Authority. 2020. CIRA-CIC-DoHBrw-2020. https://www.unb.ca/cic/datasets/dohbrw-2020.html. Accessed: 2021-10-19.

[27] Gaetano Carlucci, Luca De Cicco, and Saverio Mascolo. 2015. HTTP over UDP: An Experimental Investigation of QUIC. In *Proc. ACM SAC*. Salamanca, Spain.

[28] Catalin Cimpanu at ZDNet. 2020. Iranian Hacker Group Becomes First Known APT to Weaponize DNS-over-HTTPS (DoH). https://zd.net/3EBD8OS. Accessed: 2021-11-01.

[29] Rishabh Chhabra, Paul Murley, Deepak Kumar, Michael Bailey, and Gang Wang. 2021. Measuring DNS-over-HTTPS Performance around the World. In *Proc. ACM IMC*. Virtual Event.

[30] Catalin Cimpanu. 2019. DNS-over-HTTPS Causes More Problems than It Solves, Experts Say . https://www.zdnet.com/article/dns-over-https-causes-more-problems-than-it-solves-experts-say/. Accessed: 2021-10-11.

[31] Cisco. 2021. Alarm Category: Data Exfiltration. https://cisco.bravais.com/s/169ZRg2tWIOXb0uBjfWX. Accessed: 2021-10-12.

[32] Levente Csikor, Himanshu Singh, Min Suk Kang, and Dinil Mon Divakaran. 2021. Privacy of DNS-over-HTTPS: Requiem for a Dream? (Sep 2021).

[33] Selena Deckelmann. 2020. Firefox Continues Push to Bring DNS over HTTPS by Default for US Users. https://blog.mozilla.org/en/products/firefox/firefox-continues-push-to-bring-dns-over-https-by-default-for-us-users/. Accessed: 2021-10-21.

[34] M. Dempsky. 2010. *DNSCurve: Link-Level Security for the Domain Name System draft-dempsky-dnscurve-01*. RFC Draft. https://datatracker.ietf.org/doc/html/draft-dempsky-dnscurve-01

[35] T. Dierks and E. Rescorla. 2008. *The Transport Layer Security (TLS) Protocol*. RFC 5246. https://datatracker.ietf.org/doc/html/rfc5246

[36] Christian J. Dietrich, Christian Rossow, Felix C. Freiling, Herbert Bos, Maarten van Steen, and Norbert Pohlmann. 2011. On Botnets That Use DNS for Command and Control. In *Proc. IEEE EC2ND*. Gothenburg, Sweden.

[37] Trinh Viet Doan, Irina Tsareva, and Vaibhav Bajpai. 2021. Measuring DNS over TLS from the Edge: Adoption, Reliability, and Response Times. In *Proc. PAM*. Virtual.

[38] Dennis Fisher. 2020. Google Makes DNS Over HTTPS Default In Chrome. https://duo.sc/30z9CLm. Accessed: 2021-10-21.

[39] Sebastián García, Karel Hynek, Dmtrii Vekshin, Tomáš Čejka, and Armin Wasicek. 2021. Large Scale Measurement on the Adoption of Encrypted DNS. arXiv:2107.04436 [cs.CR]

[40] Hassan Habibi Gharakheili and Vijay Sivaraman. 2017. Cloud Assisted Home Networks. In *Proc. CAN*. Incheon, Republic of Korea.

[41] Alessandro Ghedini. 2019. Even Faster Connection Establishment with QUIC 0-RTT Resumption. https://bit.ly/3qKgsby. Accessed: 2021-10-22.

[42] Google. 2021. Google Public DNS. https://dns.google/. Accessed: 2021-12-21.

[43] Martin Grill, Ivan Nikolaev, Veronica Valeros, and Martin Rehak. 2015. Detecting DGA malware using NetFlow. In *Proc. IFIP/IEEE IM*. Ottawa, Canada.

[44] David A E Haddon and Haider Alkhateeb. 2019. Investigating Data Exfiltration in DNS Over HTTPS Queries. In *Proc. IEEE ICGS3*. London, UK.

[45] Cristian Hesselman, Merike Kaeo, Lyman Chapin, Kimberly Claffy, Mark Seiden, Danny McPherson, Dave Piscitello, Andrew McConachie, Tim April, Jacques Latour, and Rod Rasmussen. 2020. The DNS in IoT: Opportunities, Risks, and Challenges. *IEEE Internet Computing* 24, 4 (2020), 23–32.

[46] P. Hoffman and P. McManus. 2018. *DNS Queries over HTTPS (DoH)*. RFC 8484. https://tools.ietf.org/html/rfc8484

[47] Austin Hounsel, Kevin Borgolte, Paul Schmitt, Jordan Holland, and Nick Feamster. 2019. Analyzing the Costs (and Benefits) of DNS, DoT, and DoH for the Modern Web. In *Proc. ANRW*. Montreal, Quebec, Canada.

[48] Austin Hounsel, Kevin Borgolte, Paul Schmitt, Jordan Holland, and Nick Feamster. 2020. Comparing the Effects of DNS, DoT, and DoH on Web Performance. In *Proc. ACM WWW*. Taipei, Taiwan.

[49] Austin Hounsel, Paul Schmitt, Kevin Borgolte, and Nick Feamster. 2021. Can Encrypted DNS Be Fast?. In *Proc. PAM*. Virtual Event.

[50] Austin Hounsel, Paul Schmitt, Kevin Borgolte, and Nick Feamster. 2021. Encryption without Centralization: Distributing DNS Queries across Recursive Resolvers. In *Proc. ANRW*. Virtual Event, USA.

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

[52] Xin Hu, Jiyong Jang, Marc Ph. Stoecklin, Ting Wang, Douglas L. Schales, Dhilung Kirat, and Josyula R. Rao. 2016. BAYWATCH: Robust Beaconing Detection to Identify Infected Hosts in Large-Scale Enterprise Networks. In *Proc. IEEE/IFIP DSN*. Portland, Oregon, USA.

[53] Z. Hu, L. Zhu, J. Heidemann, A. Mankin, D. Wessels, and P.Hoffman. 2016. *Specification for DNS over Transport Layer Security (TLS)*. RFC 7858. https://tools.ietf.org/html/rfc7858

[54] Qing Huang, Deliang Chang, and Zhou Li. 2020. A Comprehensive Study of DNS-over-HTTPS Downgrade Attack. In *Proc. USENIX FOCI*.

[55] C. Huitema, S. Dickinson, and A. Mankin. 2021. *Specification of DNS over Dedicated QUIC Connections*. RFC. https://datatracker.ietf.org/doc/html/draft-ietf-dprive-dnsoquic-04

[56] Karel Hynek and Tomas Cejka. 2020. Privacy Illusion: Beware of Unpadded DoH. In *Proc. IEEE IEMCON*. Vancouver, Canada.

[57] Karel Hynek, Tomas Cejka, and Dmitrii Vekshin. 2020. DoH detection: Discovering hidden DNS. In *Proc. PESW*. Czech Republic.

[58] J. Iyengar and M. Thomson. 2021. *QUIC: A UDP-Based Multiplexed and Secure Transport*. RFC. https://datatracker.ietf.org/doc/rfc9000/

[59] Ali Sadeghi Jahromi and AbdelRahman Abdou. 2021. Comparative Analysis of DoT and HTTPS Certificate Ecosystems. In *Proc. NDSS MADWeb*.

[60] Philipp Jeitner and Haya Shulman. 2021. Injection Attacks Reloaded: Tunnelling Malicious Payloads over DNS. In *Proc. 30th USENIX Security*. Virtual Event, USA.

[61] Philipp Jeitner, Haya Shulman, and Michael Waidner. 2020. Secure Consensus Generation with Distributed DoH. In *Proc. IEEE-IFIP DSN-S*. Valencia, Spain.

[62] Philipp Jeitner, Haya Shulman, and Michael Waidner. 2020. The Impact of DNS Insecurity on Time. In *Proc. IEEE/IFIP DSN*. Valencia, Spain.

[63] Lin Jin, Shuai Hao, Haining Wang, and Chase Cotton. 2021. Understanding the Impact of Encrypted DNS on Internet Censorship. In *Proc. ACM WWW*. Ljubljana, Slovenia.

[64] R. Kartch. 2017. Best Practices for Network Border Protection. Carnegie Mellon University's Software Engineering Institute Blog. http://insights.sei.cmu.edu/blog/best-practices-for-network-border-protection/

[65] E. Kinnear, P. McManus, T. Pauly, T. Verma, and C. A. Wood. 2021. *Oblivious DNS Over HTTPS*. RFC. https://datatracker.ietf.org/doc/html/draft-pauly-dprive-oblivious-doh-07

[66] Ludmila Kudryavtseva. 2018. DNSCrypt Has Quit, But You Needn't Worry (UPDATED). https://adguard.com/pt_pt/blog/bye_dnscrypt.html. Accessed: 2021-10-12.

[67] Carmen Kwan, Paul Janiszewski, Shela Qiu, Cathy Wang, and Cecylia Bocovich. 2021. Exploring Simple Detection Techniques for DNS-over-HTTPS Tunnels. In *Proc. ACM FOCI*. Virtual Event, USA.

[68] J. Lee, J. Kwon, H. Shin, and H. Lee. 2010. Tracking Multiple C&C Botnets by Analyzing DNS Traffic. In *IEEE Workshop on Secure Network Protocols*. Kyoto Japan.

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

[70] Minzhao Lyu, Hassan Habibi Gharakheili, Craig Russell, and Vijay Sivaraman. 2021. Hierarchical Anomaly-Based Detection of Distributed DNS Attacks on Enterprise Networks. *IEEE Transactions on Network and Service Management* 18, 1 (2021).

[71] Minzhao Lyu, Hassan Habibi Gharakheili, Craig Russell, and Vijay Sivaraman. 2019. Mapping an Enterprise Network by Analyzing DNS Traffic. In *Proc. PAM*. Puerto Varas, Chile.

[72] Alexander J Martin. 2019. Google's Chrome Browser Plans 'Risk Undermining Fight Against Online Child Abuse', Govt Warned. https://bit.ly/3beOLij. Accessed: 2021-10-11.

[73] A. Mayrhofer. 2018. *Padding Policies for Extension Mechanisms for DNS*. RFC 8467. https://datatracker.ietf.org/doc/html/rfc8467

[74] Enock S. Mbewe and Josiah Chavula. 2020. On QoE Impact of DoH and DoT in Africa: Why a User's DNS Choice Matters. In *Proc. EAI AFRICOMM*. Ebene City, Mauritius.

[75] P. Mockapetris. 1987. *Domain Names - Implementation and Specification*. RFC 1035. https://datatracker.ietf.org/doc/html/rfc1035

[76] Mohammadreza MontazeriShatoori, Logan Davidson, Gurdip Kaur, and Arash Habibi Lashkari. 2020. Detection of DoH Tunnels using Time-series Classification of Encrypted Traffic. In *Proc. IEEE DASC/PiCom/CBDCom/CyberSciTech*. Online Event.

[77] Michael Mühlhauser, Henning Pridöhl, and Dominik Herrmann. 2021. Code and Datasets for the ARES 2021-Paper: How Private is Android's Private DNS Setting? Identifying Apps by Encrypted DNS Traffic. https://github.com/UBA-PSI/segram. Accessed: 2021-11-03.

[78] Michael Mühlhauser, Henning Pridöhl, and Dominik Herrmann. 2021. How Private is Android's Private DNS Setting? Identifying Apps by Encrypted DNS Traffic. In *Proc. ARES*. Vienna, Austria.

[79] Asaf Nadler, Avi Aminov, and Asaf Shabtai. 2019. Detection of Malicious and Low Throughput Data Exfiltration over the DNS Protocol. *Computers & Security* 80 (Jan 2019), 36–53.

[80] Yoshimichi Nakatsuka, Andrew Paverd, and Gene Tsudik. 2019. PDoT: Private DNS-over-TLS with TEE Support. In *Proc. ACSAC*. San Juan, Puerto Rico, USA.

[81] Yoshimichi Nakatsuka, Andrew Paverd, and Gene Tsudik. 2021. PDoT: Private DNS-over-TLS with TEE Support. *Digital Threats: Research and Practice* (Feb 2021).

[82] Gopi Nath Nayak and Shefalika Ghosh Samaddar. 2010. Different Flavours of Man-In-The-Middle Attack, Consequences and Feasible Solutions. In *Proc. IEEE ICCSIT*. Chengdu, China.

[83] Network Security Research Lab at 360. 2019. An Analysis of Godlua Backdoor. https://bit.ly/2YSqyf4. Accessed: 2021-10-29.

[84] Tomasz Andrzej Nideck. 2019. DoH: Mozilla, Cloudflare, and Google vs. the World. https://bit.ly/3cjsBMl. Accessed: 2021-10-24.

[85] Alexandra Nisenoff, Nick Feamster, Madeleine A Hoofnagle, and Sydney Zink. 2021. User Expectations and Understanding of Encrypted DNS Settings. In *Proc. NDSS DNS Privacy Workshop*. Virtual Event.

[86] Palo Alto Networks. 2021. Command and Control Explained. https://www.paloaltonetworks.com/cyberpedia/command-and-control-explained. Accessed: 2021-10-22.

[87] Constantinos Patsakis, Fran Casino, and Vasilios Katos. 2020. Encrypted and Covert DNS Queries for Botnets: Challenges and Countermeasures. *Computers & Security* (Jan 2020).

[88] Kira Radinsky. 2015. Data Monopolists Like Google Are Threatening the Economy. Harvard Business Review. https://bit.ly/3yT6dEb

[89] Sean Rivera, Vijay K. Gurbani, Sofiane Lagraa, Antonio Ken Iannillo, and Radu State. 2020. Leveraging EBPF to Preserve User Privacy for DNS, DoT, and DoH Queries. In *Proc. ARES*. Virtual Event, Ireland.

[90] Bushra Sabir, Faheem Ullah, M. Ali Babar, and Raj Gaire. 2021. Machine Learning for Detecting Data Exfiltration: A Review. *ACM Comput. Surv.* 54, 3 (May 2021).

[91] Paul Schmitt, Anne Edmundson, Allison Mankin, and Nick Feamster. 2019. Oblivious DNS: Practical Privacy for DNS Queries: Published in PoPETS 2019. In *Proc. ANRW*. Montreal, Quebec, Canada.

[92] Michael Sehring. 2020. Securing DNS Traffic: An Introduction to DoT & DoH. https://bit.ly/33T5AyX. Accessed: 2021-10-22.

[93] Dongwan Shin and Rodrigo Lopes. 2011. An Empirical Study of Visual Security Cues to Prevent the SSLstripping Attack. In *Proc. ACSAC*. Orlando, Florida, USA.

[94] Haya Shulman. 2014. Pretty Bad Privacy: Pitfalls of DNS Encryption. In *Proc. ACM WPES*. Scottsdale, Arizona, USA.

[95] Sandra Siby, Marc Juarez, Narseo Vallina-Rodriguez, and Carmela Troncoso. 2018. DNS Privacy not So Private: the Traffic Analysis Perspective. In *Proc. HotPETs*. Barcelona, Spain.

[96] Sandra Deepthy Siby, Marc Juárez, Claudia Díaz, Narseo Vallina-Rodriguez, and Carmela Troncoso. 2020. Encrypted DNS → Privacy? A Traffic Analysis Perspective. In *Proc. NDSS*. San Diego CA, USA.

[97] Manmeet Singh, Maninder Singh, and Sanmeet Kaur. 2019. Detecting Bot-Infected Machines using DNS Fingerprinting. *Digital Investigation* 28 (2019), 14–33.

[98] Sunil Kumar Singh and Pradeep Kumar Roy. 2020. Detecting Malicious DNS over HTTPS Traffic Using Machine Learning. In *Proc. 3ICT*. Virtual Event.

[99] Daniel Stenberg. 2014. HTTP2 Explained. *ACM SIGCOMM Computer Communication Review* 44 (Jul 2014).

[100] The Proofpoint Threat Insight Team. 2019. PsiXBot Now Using Google DNS over HTTPS and Possible New Sexploitation Module. https://bit.ly/3Eri5Pb. Accessed: 2021-10-29.

[101] Sadegh Torabi, Amine Boukhtouta, Chadi Assi, and Mourad Debbabi. 2018. Detecting Internet Abuse by Analyzing Passive DNS Traffic: A Survey of Implemented Systems. *IEEE Communications Surveys & Tutorials* 20, 4 (2018).

[102] Agung Udiyono, Charles Lim, and Lukas. 2020. Botnet Detection Using DNS and HTTP Traffic Analysis. In *Proc. ACM ICONETSI*. Tangerang, Indonesia.

[103] Faheem Ullah, Matthew Edwards, Rajiv Ramdhany, Ruzanna Chitchyan, M. Ali Babar, and Awais Rashid. 2018. Data Exfiltration: A Review of External Attack Vectors and Countermeasures. *Journal of Network and Computer Applications* 101 (2018), 18–54.

[104] N. Usman Aijaz, Mohammed Misbahuddin, and Syed Raziuddin. 2020. Survey on DNS-Specific Security Issues and Solution Approaches. In *Data Science and Security*.

[105] Marshall Vale and Alexander Dupuy. 2019. Google Public DNS over HTTPS (DoH) Supports RFC 8484 Standard. https://bit.ly/2YSqyvA. Accessed: 2021-10-21.

[106] Gaurav Varshney, Padmavathi Iyer, Pradeep Atrey, and Manoj Misra. 2021. Evading DoH via Live Memory Forensics for Phishing Detection and Content Filtering. In *Proc. COMSNETS*. Virtual Event.

[107] Dmitrii Vekshin, Karel Hynek, and Tomas Cejka. 2020. *Dataset used for detecting DNS over HTTPS by Machine Learning*. https://doi.org/10.5281/zenodo.3906526

[108] Dmitrii Vekshin, Karel Hynek, and Tomas Cejka. 2020. DoH Insight: Detecting DNS over HTTPS by Machine Learning. In *Proc. ARES*. Virtual Event, Ireland.

[109] Yue Wang, Anmin Zhou, Shan Liao, Rongfeng Zheng, Rong Hu, and Lei Zhang. 2021. A Comprehensive Survey on DNS Tunnel Detection. *Computer Networks* 197 (2021), 108322.

[110] Maria C. Wasastjerna. 2018. The Role of Big Data and Digital Privacy in Merger Review. *European Competition Journal* 14, 2-3 (2018), 417–444.

[111] Peter Wu. 2019. DNS Encryption Explained. https://blog.cloudflare.com/dns-encryption-explained/. Accessed: 2021-10-11.

[112] Kui Xu, Patrick Butler, Sudip Saha, and Danfeng (Daphne) Yao. 2013. DNS for Massive-Scale Command and Control. *IEEE Trans. Dependable Secur. Comput.* 10, 3 (May 2013).

[113] Zhiwei Yan and Jong-Hyouk Lee. 2020. The Road to DNS Privacy. *Future Generation Computer Systems* 112 (2020), 604–611.

[114] Dan York. 2010. CHAPTER 3 - Eavesdropping and Modification. In *Seven Deadliest Unified Communications Attacks*, Dan York (Ed.). Syngress, Boston, 41–69.

[115] Yuwei Zeng, Xiaochun Yun, Xunxun Chen, Boquan Li, Haiwei Tsang, Yipeng Wang, Tianning Zang, and Yongzheng Zhang. 2021. Finding Disposable Domain Names: A Linguistics-based Stacking Approach. *Computer Networks* 184 (Jan 2021).

[116] Yury Zhauniarovich, Issa Khalil, Ting Yu, and Marc Dacier. 2018. A Survey on Malicious Domains Detection through DNS Data Analysis. *ACM Comput. Surv.* 51, 4 (July 2018).

[117] Liang Zhu, Zi Hu, John Heidemann, Duane Wessels, Allison Mankin, and Nikita Somaiya. 2015. Connection-Oriented DNS to Improve Privacy and Security. In *Proc. IEEE S&P*. San Jose, CA, USA.