JOURNAL OF INFORMATION SCIENCE AND ENGINEERING XX, XXXX-XXXX (2024)
DOI: XX.XXX.XXXX.XX

A DNS Threat Awareness Practical Framework Using Knowledge Graph

ABDULHADI ALBLUWI, UMAR ALBALAWI+ AND ABDELRAHMAN OSMAN ELFAKI
College of Computing and Information Technology
University of Tabuk
Tabuk, Saudi Arabia 71491
E-mail: abdulhadi@aloda.org; ualbalwi@ut.edu.sa; a.efaki@ut.edu.sa

Currently, addressing DNS systems has become a daily activity for many people. People's reliance on DNS systems has attracted attention for DNS cybersecurity threats. In this paper, a DNS threat awareness practical framework is presented. As a methodology, first, the main DNS threats are classified, technically explained, and illustrated by examples that show how a hacker uses a specific DNS threat and creates vulnerabilities. Second, related works that have addressed DNS threats in the last 5 years are investigated to highlight the research gap in this field. Third, methods for preventing DNS threats are selected based on their reputation in industry. Experiments are conducted to prove the applicability of the selected DNS threat prevention methods, and to identify their pros and cons. A technical awareness framework in the form of usage guidance is generated based on these pros and cons and is considered as the main contribution of this paper. Moreover, the proposed framework provides recommendations to improve the privacy, security, and performance of DNS resolution from the client perspective. A security triad (confidentiality, integrity, and availability) is chosen as a benchmark for evaluating the proposed framework. The contributions of this paper could be summarized as: providing a comprehensive analysis of DNS threats, which highlights a clear understanding of each DNS threats, providing a dynamic framework as usage guidance to defeat DNS threats provided with rich database collected from CVE, CWE, CAPEC, and CPE provided by MITRE and NIST and connects them using Neo4j: which propose a knowledge graph for DNS threats. This knowledge graph represents the result of creating a knowledge representation that can effectively combine data from as many sources as possible within the cybersecurity domain which in our case is DNS threats awareness. To the best of our knowledge this is the first paper to provide such framework that supports DNS-over-Encryption protocols, developing a benchmark based on the basic security triad (confidentiality, integrity, and availability) for comparing DNS threat prevention methods, and finally, developing recommendations to improve the privacy, security, and performance of DNS resolution from the client perspective.

Keywords: cyber-security, DNS over encryption, security policies, DNS threats awareness.

1. INTRODUCTION

Modern life has become mainly dependent on the internet in all its aspects. In entertainment, television platforms have become an alternative to regular television stations,

Received .....; revised ....; accepted ......

+Corresponding author Umar Albalawi

digital media files on the internet have become an alternative to CDs, and electronic games on the internet have become a popular alternative to regular games or even games that work offline. In the field of education, electronic learning platforms have become integral to the process of communication between teachers and students, and online courses have successfully addressed the rapid development in various scientific fields. As a general knowledge, commercial works have become fundamentally dependent on the internet in terms of identifying new markets and resources and creating new partnerships. Hence, it has become almost impossible to complete any kind of business without relying on the internet. Therefore, we can say, DNS is the protocol that makes the internet work by allowing users to reach their requested websites.

Websites are considered the mainstay of the internet, where each website is defined by using its Domain Name System (DNS). The DNS is defined as a system used to recognize computers that are reachable via the internet [1]. Since DNS is commonly transported over UDP/IP, it is easy for any attacker to generate packets that comply with the transport protocol parameters. The resource records contained in the DNS associate domain names with other forms of information.

Al-Mashhadi [2] proved that DNS traffic detection is one of the necessary factors of botnet communication attacks. DNS packets are widely trusted by firewalls and cybersecurity defender software; thus, DNS security [3]. In general, there are two DNS threads: DNS hijack and DNS leak [4, 5]. In a DNS hijack, a user believes that they are connecting to a legitimate domain while they are actually traffic is completely allowed to pass freely through the cybersecurity defender software. However, DNS traffic is commonly attacked and abused by cybercriminals. Therefore, the security of a DNS is a significant component of the network connecting to a malicious domain. In a DNS leak, the DNS query is exposed, which leads to the extraction of personal information such as the recipient's and sender's IP, location, and web searches.

The DNS is a critical part of Internet communication, and it plays a crucial role on the internet, in addition, spread of IoT applications over the world made demands of Internet applications lead to huge increasing in DNS sites. This revolution in DNS site attracts hackers and cybercriminal to create new or try well known DNS spoofing methods for it been an unencrypted protocol over UDP (a connectionless protocol) or over TCP port 53, making it easy to intercept traffic with spoofing. Furthermore, DNS servers do not provide validation of the IP addresses to which they are redirecting traffic to. For these reasons, many solutions introduced recently to overcome these vulnerabilities of DNS, over time, multiple solution and tools have been developed to address these challenges. Despite this effort, not all challenges were addressed, nor one solution fit the needs of CIA security Triad or vulnerabilities.

This study aims to deal with DNS spoofing. In order to address the identified problems, the following research questions are formulated:

- Q1: What are the methods to prevent the DNS spoofing (cache poisoning) and the features of these methods? Illustrated in Section 3.2.

- Q2: What is the proper prevention method for every case from the client (end-users) prospective? Explained in Section 4.

- Q3: How to enhance awareness of DNS security? Proposed in section 5.

The last question is considering as our main research question. This paper is interested in providing answers to the above research questions. The following steps are designed to address the research questions and break down the tasks into specific clarifications: (1) classify and identify the main features of DNS threats; (2) investigate the DNS spoofing prevention methods; (3) develop practical framework to secure DNS.

2. RELATED WORKS

Moubayed [6] proposed a machine learning-based approach to address the typo squatting vulnerability. Their approach detects suspicious domains with high accuracy. Moreover, the observed trends are validated by analyzing the same features in an unlabeled dataset using the K-means clustering algorithm. The results show that legitimate domains have a shorter domain name length and fewer unique characters. Moreover, the developed ensemble learning classifier performs better in terms of accuracy, precision, and F-score. However, the number of domains identified as potentially suspicious is high. Hence, the ensemble learning classifier is applied with results showing that the number of domains identified as potentially suspicious is reduced by almost a factor of five while still maintaining the same trends in terms of feature statistics. Hananto [7] introduced a method to detect denial of service attacks by using NetFlow traffic that indicates DDOS attacks and DNS traffic early to validate DNS DDOS attacks. By measuring the statistical entropy of NetFlow traffic and the statistical values of the DNS NXDOMAIN response, the model can be used to detect either low-volume denial-of-service attacks or high-volume denial-of-service attacks. Spaulding [8] presented a technique for proactive detection algorithm generated malicious domain names employed by botnets. The authors devised a detection algorithm using the notion of the difference function over the number of NXDomain responses for a given domain with a sliding time window. Using DNS traffic gathered from certain TLDs for the precalculated list of generated domains by the Conficker malware variants, their detection algorithm was able to achieve 99% accuracy as early as 48 hours prior to registration.

Chau [9] proposed CGuard, an adaptive defense framework in which CGuard actively detects cache poisoning, attempts and protects the cache entries under attack by exclusively updating them via available high secure channels. Mittal [10] presented a novel distributive denial of service attack prevention mechanism by utilizing the flexibility and programmability aspects of software-defined networks (SDNs). The premise of the mechanism is to route DNS response packets along the same path that was utilized by the corresponding DNS request packet. This way, the malicious host responsible for launching a distributive denial of service attack will self-destruct. There is a disadvantage in their proposed paper: an additional delay of 8%–9% in obtaining DNS responses compared to the current DNS structure. Almusawi and Amintoosi [11] introduced a solution that detects and classifies DNS tunneling. The experimental results demonstrate the efficacy of the proposed SVM classification method by obtaining a measure of 0.80. Deccio [12] conducted experiments to test DNS security by issuing recursive DNS queries to a large group of servers using various spoofed addresses. The authors tested half of 62,000 networks in which 4.6% received the request and addressed it using a reliable repeat inquiry. A total of 6.2% networks addressed the scam source requests, whereas the remaining networks were exposed to scam sources. The authors identified approximately 4,000 DNS server instances at risk of cache poisoning attacks.

Bumanglag [13] conducted experiments for testing the DNS over HTTP (DoH). The results proved that the DoH improves confidentiality, but on the other hand, the DOH can be used by malware. Therefore, malicious activity should be detected first. Majundar [14] proposed ARP poisoning and detection tools built and demonstrated via simple tools using Scapy, which enable the user to attack and detect ARP spoofing. A script written using the Scapy library is selected to obtain the response MAC address from the sniffed ARP Packet received by the target machine. Setting up a Static ARP entry in the ARP cache table sets up a permanent entry in the ARP cache. This entry is used as a protection layer against ARP spoofing attacks. Setting up a static entry for an address prevents the device from listening to ARP responses for that address, and thus, ARP spoofing is prevented since the ARP cache of the target cannot be altered for the said MAC address. Static entries for all frequently utilized entries are set up in the ARP Cache table. Any attempt to change the entry in the cache table will disconnect the IP address of the attacker from the network. The limitation of [14] work is focusing on static ARP entries as the prevention method for ARP spoofing. Using static ARP entries is recommended for smaller networks, as it necessitates immense administrative overhead. A static ARP entry is added for each system on the network to every individual system. Hynek [15] investigated DoH threats and defined three reasons for DoH abuse: system command, changing the target channel, and oblivious usage. Nadler [16] conducted experiments to review countermeasures against DNS privacy leakage. According to their results, the DoH is the most applicable solution and prevents practical attacks on the DNS protocol and its applications. Singh [17] developed a machine learning model to test the applicability of the DoH and employed a DNS over the HTTPS dataset. The results of this learning model confirm that the DNS over HTTPS is the best choice to achieve security because this model detects the most malicious activities. Bumanglag and Kettani [18] analyzed DDoS attacks over the DNS. Their study classified DDoS and predicted the severity and mitigation of DDoS attacks. Table 1 shows summary of related works in DNS security in the last five years. The aforementioned analysis of related works proves the following facts:

- For each DNS threat, there is more than one solution that depends on the environment and application type.
- Any successful DNS threat prevention method needs to be proven by intense experience. Using mathematical models is insufficient for proving the success of the DNS prevention method.

In of the previous facts, the research gap could be a critical need for providing guidance for security professionals in applying a suitable DNS threat prevention solution.

3. MATERIALS AND METHODS

In this section, the detailed steps of a roadmap for implementing the proposed framework are presented as shown in Fig. 1. The following steps are listed:

Table 1. Summary of related works in DNS security.

Work	Problem	Solution
[6] (2018)	Typo squatting vulnerability	Machine learning model to discover DNS typosquatting
[7] (2018)	DNS DDoS attack	Detect DDoS attack by measuring statistical entropy of NetFlow traffic
[8] (2018)	DGA domains	Devised a detection algorithm
[9] (2018)	Cache poisoning	Detect cache poisoning attempts and protect the cache entries
[10] (2018)	DDoS attack	Prevention mechanism
[11] (2018)	DNS tunneling	Detecting DNS tunneling
[12] (2020)	Cache pointing attack	Identifying and testing DNS security
[13] (2020)	DNS over HTTPS	Prevention DNS threats by DoH
[14] (2021)	ARP Poisoning	Preventing ARP spoofing by Static IP table
[15] (2022)	DoH abuse	No solution is provided
[16] (2022)	Information disclosure	Prevention DNS threats by DoH
[17] (2022)	DDoS attack	DoH
[18] (2022)	DDoS attack	No solution is provided

- Investigation the literature to extract the main threats that could be caused by attacking DNS, i.e., common threats of DNS infrastructure, for each threat, an example with a discussion is provided. In addition, the associated DNS records are defined.

- The methods for preventing DNS threats are defined, and a comprehensive analysis of each method is provided.

- A DNS awareness framework is developed by considering the extracted features from the previous step. Table 2 shows the logical components of the proposed methodology.

Table 2. The logic components of the proposed method.

Input	Methodology	Output
Related works	Investigate and analysis of the related works	Defining the main DNS threats
DNS prevention methods	Analyze DNS prevention methods	Features of each DNS prevention methods
Features of DNS prevention methods	Evaluation of DNS prevention methods	Prove applicability of the DNS prevention methods
Experiment [19] results	Develop framework to enhance DNS security	Framework
Framework	Evaluations of the developed framework	Evaluations Results

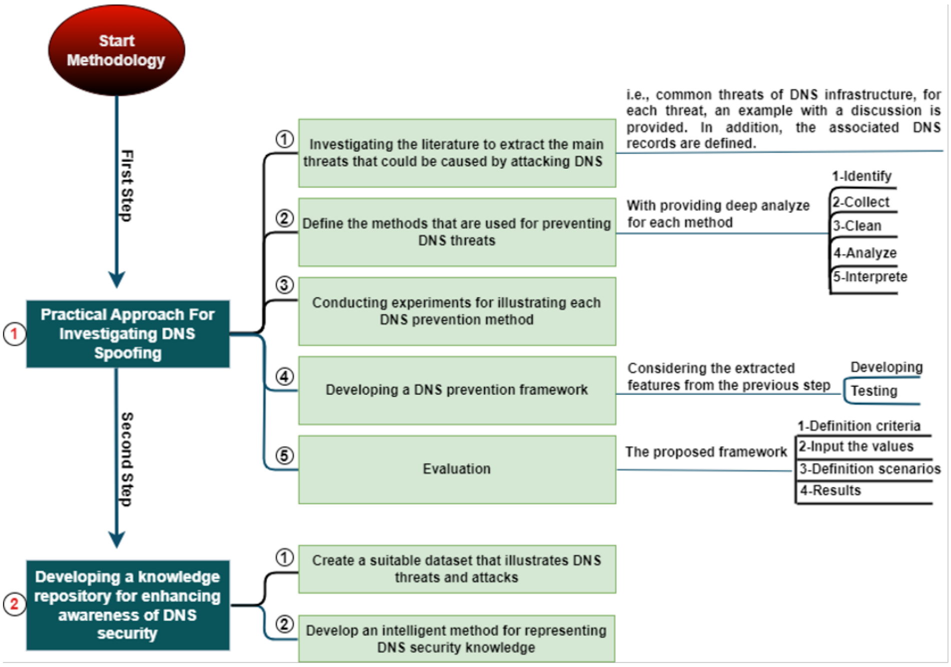

Start
Methodology
First Step
1
Practical Approach For Investigating DNS Spoofing
Second Step
2
Developing a knowledge repository for enhancing awareness of DNS security
①
Investigating the literature to extract the main threats that could be caused by attacking DNS
i.e., common threats of DNS infrastructure, for each threat, an example with a discussion is provided. In addition, the associated DNS records are defined.
②
Define the methods that are used for preventing DNS threats
With providing deep analyze for each method
1-Identify
2-Collect
3-Clean
4-Analyze
5-Interprete
③
Conducting experiments for illustrating each DNS prevention method
④
Developing a DNS prevention framework
Considering the extracted features from the previous step
Developing
Testing
⑤
Evaluation
The proposed framework
1-Definition criteria
2-Input the values
3-Definition scenarios
4-Results
①
Create a suitable dataset that illustrates DNS threats and attacks
②
Develop an intelligent method for representing DNS security knowledge

Fig. 1. The steps of the proposed methodology.

3.1 Common threats of DNS infrastructure

This section presents the first step in our proposed methodology, which is to investigate the literature to extract the main threats that could be caused by attacking the DNS. In this step, we extracted eleven threats, which are illustrated in the following section.

3.1.1 DNS cache poisoning (DNS Spoofing)

The process of injecting false DNS records into a server's cache is known as cache poisoning or DNS spoofing. A common method for DNS Spoofing is to respond to an information request with false data that has a spoofed the source IP address. The inaccurate information might be cached if the fake information reply arrives before the requested server's initial response. Any subsequent information requests will receive this false information until the information expires once the cache has been poisoned with false information in this manner. Therefore, local users will receive the contaminated information if a corporate resolver is compromised. The consequences will be much more severe if a public resolver or the resolver for an ISP is poisoned [20]. Below, an illustrated example is presented.

Example 1: If example.bank.net is at 192.168.3.13 and evil.net is at 198.168.4.13, by sending a forged response to a DNS server, an attacker may attempt to poison its cache. example.bank.net is at 198.168.4.13.

3.1.2 Kaminsky

Dan Kaminsky's approach is more effective than DNS Cache poisoning, where the key difference is the nature of the forged payload: we can go up one level and hijack the authoritative server's records instead. In conclusion, Dan Kaminsky discovered a critical flaw that affected major DNS server vendors and gave an attacker more control over their attempts to poison a recursive server's cache. This flaw allows the attacker to:

- Launch multiple DNS queries that are probably not already in a cache to cause recursion.
- Poison the target name's cache as well as any other record in that domain by using the target name as a referral [21]. Another type of DNS cache poisoning is ID guessing and query prediction.

3.1.3 ID guessing and query prediction

There are only 2^{32} possible combinations of ID and client UDP ports for a given client and server because the ID field in the DNS header is only a 16-bit field and the server UDP port associated with the DNS is a well-known value. This is not a very wide range, so it is insufficient to thwart a brute force search. In addition, the client port is frequently a known fixed value because of firewalls or other restrictions, which further reduces the search space to a range smaller than 216. In practice, both the client UDP port and the ID can frequently be predicted from previous traffic. When used separately, ID guessing injection of bogus responses. Since this attack relies on predicting a resolver's behavior, it is most likely to be successful when the victim is in a known state, because the victim either recently rebooted or the victim's behavior has been affected by the attacker's other actions or by the victim's predictable response to a third party action known to the attacker [22]. The third type of DNS cache poisoning is known as name chaining.

3.1.4 Name chaining

Attackers inject responses, whether by packet interception, guessing the victim's query, or posing as a legitimate name server and participating in the victim's query response at some point. One or more RRs with DNS names in their RDATA are included in the attacker's response. Depending on the specific form of this attack, the goal may be either to inject false data associated with those names into the victim's cache in the additional section of this response or to redirect the next stage of the query to a server of the attacker's choosing (putting the lies in the Authority or Answer section of the response, where they have a better chance of evading a resolver's defenses, or injecting more complex lies into the victim's cache than will fit comfortably in a single response). There are cache poisoning attacks that are not named chaining attacks in the sense discussed here because any attacker who can insert resource records into a victim's cache can almost certainly cause harm. Name chaining attacks, however, deserve special consideration because the cause-and-effect relationship between the initial attack and the ultimate result may be significantly more complex than in the case of other types of cache poisoning. In all name chaining attacks, the attacker introduces arbitrary DNS names of their choosing and provides additional information that they claim is associated with those names. The victim will find it difficult to defend against this class of attacks unless they have a better understanding of the data associated with those names. Given how simple it is for an attacker to provoke a victim into asking for a specific name of the attacker's choosing, this type of attack is especially sneaky [23]. One method would be to include a link to a $1x1$-pixel ”web bug” graphic in a text/HTML message sent to the victim. Whenever the victim's mail reading program attempts to follow such a link, the result will be a DNS query for a name chosen by the attacker.

3.1.5 Amplification/reflector DNS (Dos/DDos attack)

Reflection/amplification attacks are a common form of distributed denial of service (DDoS) attacks that target DNS servers. In this type of attack, the victims' internet Protocol (IP) addresses are employed to create spoofed DNS request packets. Since the DNS uses the User Datagram Protocol (UDP), which is a connectionless transport layer protocol that lacks any handshaking mechanisms or techniques, rather than responding to the packet's source, the DNS server instead addresses the victim. Amplification of the response occurs as a result of the attacker's search for response types that are many times larger than the corresponding request. The amplification factor (AF) is the ratio of the size of the response to the size of the request.

According to Anagnostopoulos [24], one dangerous DDoS that occurred in 2013 was distributing spoofed DNS requests to numerous open resolvers. The open resolver is defined as a DNS server that recursively addresses random DNS queries from anonymous sources in cyberspace. As a result of this spoofing, all DNS responses are redirected to that hacker website. Furthermore, introducing the DNS Security Extension (DNSSEC) amplification factor is significantly increased by increasing the size of the DNS responses from 512 bytes to 4096 bytes. DNSSEC is crucial for defending DNS servers against assaults such as cache poisoning. However, when an attacker uses DNSSEC to quickly generate more traffic, it makes their job easier [25].

3.1.6 Botnet DNS attacks

A botnet attack has been defined as a collection of digital devices or computers that have been infected with malware to give hackers control over them. Cybercriminals use botnets to launch attacks such as DDoS attacks, credential leaks, data theft, and unauthorized access to information. One should be able to recognize network attacks to reduce such security risks [26]. In some internet protocols, botnets hide their malicious activities and evade detection.

3.1.7 DNS manipulation

DNS manipulation is the process of routing legitimate DNS requests to erroneous IP addresses that are kept on unreliable servers. Users are exposed to risks such as phishing and content injection due to attackers' DNS manipulation behavior [27]. Schomp [28] assessed the susceptibility of user-side DNS infrastructure to record injection threats. According to their analysis, 9% of open DNS resolvers are susceptible to record injection attacks and are inappropriately utilized by attacks against shared DNS infrastructure. The authors have assessed the severity of well-known record injection attacks such as Kaminsky [29] as well as the use of well-known defensive techniques such as $0x20$ encoding [30]. Kührer [31] explored the drawbacks of open DNS resolvers, which can be maliciously used by attackers for a variety of bad deeds such as DDoS attack amplification, DNS manipulation, and cache poisoning. The authors examined the open resolvers' responses for authenticity from the user's perspective and discovered that millions of them purposefully alter DNS resolutions to censor communication channels, injecting advertisements, serving malicious files, and performing phishing attacks.

3.1.8 Malicious domains

Malicious domains are crucial to many attack strategies. Malicious domains are crucial to the success of almost all well-known attack vectors, from disseminating malware to housing command and control (CC) servers and traffic distribution [32]. For instance, domain names have become more frequently utilized by attackers, who gain the ability to modify the IP address of the malicious servers they control by using the DNS. To make their malicious servers more challenging to locate and take down, they can also conceal their vital servers behind proxy services, such as FastFlux [33]. Attackers have the freedom to easily migrate their malicious servers with the use of domain names. The malicious ”services” that the attackers provide multiply and are “fault-tolerant” with respect to the IP addresses where they are hosted.

3.1.9 Domain generation algorithm(DGA)

According to Schiavoni [34], the DGA is a dynamic method of communicating with a centralized server to Malware frequently employs the DGA, which is a sequencing algorithm, to create numerous domain names on a regular basis to circumvent domain-based firewall protection. The malicious actors have the chance to conceal their C2 servers using the generated domain names, making it difficult for the enterprise to recognize the DGA. The domains created by DGAs are short-lived, registered domains that are simpler for humans to recognize than in automatic machine detection.

3.1.10 Domain name squatting

Domin name squatting is the practice of registering or using an internet domain name with the intention of making money off of another person's trademark. By a variety of squatting techniques, as listed below, cybersquatters register variations of well-known trademark names:

- Typosquatting: This practice focuses on internet users who incorrectly enter a website address, such as www.examlpe.com rather than www.example.com, and is primarily employed for monetization purposes [35].

- Bitsquatting is a term used to describe a hardware issue that involves registering domain names with one different bit from well-known domain names to capture unintentional traffic generated by bit-flip errors in computer memory [36].

- Combosquatting: A well-known brand name and one or more phrases, such as youtube-login.com in place of youtube.com, are utilized. Kintis [37] investigated its reach.

- Soundsquatting: This is a method for creating squatted domains that takes advantage of homophones and the user's confusion. Examples include weather, idle, idol, and idyll [38].

Users are vulnerable to many threats as a result of domain name squatting practices, including malware, scams, monetization, and trademark infringement. Table3 shows examples of several types of domain name squatting for the website (gmail[.]com).

Table 3. Example of several types of domain name squatting for the gmail[.]com.

gmail[.]com	Original Domain
gmaill[.]com	Typosquatting
jmail[.]com	Soundsquatting
gmailg[.]com	Bitsquatting
gmail-login[.]com	Combosquatting

3.1.11 Privacy leakage

Privacy leakage happens when the resolver (DNS provider) logs the IP addresses of its users and the domain names in which they are interested or learn informational queries of that domain. If the DNS provider learns the desired domain, privacy may be at risk. For instance, it is assumed that the adversary has the ability to compare the popularity of any two domain names that are registered with a specific registrar. When the owner of a well-known domain name forgets to renew it, the adversary could then take it [38].

3.2 Methods for preventing DNS threats

In this section, we focus on DNS threat prevention methods. Therefore, traditional intrusion detection methods such as firewall and network intrusion detection algorithms are disregarded. In this section, four DNS threat prevention solutions are discussed: DNS over TLS (DoT), DNS over HTTPS (DoH), DNS over QUIC (DoQ), and DNSSEC. The first three methods are defined as encrypted DNS protocols (DoE), and the fourth method is defined as digital signatures. We evaluate those proposed prevention solutions and techniques by specific criteria relying on the security triad, Confidentiality, Integrity, and Availability (CIA). In the following section, the pros and cons of every method are presented in detail.

3.2.1 DNS-over-TLS (DoT)

DNS TLS, or DoT, is standardized by RFC7858 and is a standard for encrypting DNS queries, hence, preventing passive monitoring. Resolvers use authentication by verifying SSL certificates, hence, preventing man-in-the-middle attacks. DoT uses the same encryption and authentication protocol, TLS, as HTTPS websites do. (SSL is another name for TLS.) The user datagram protocol (UDP), which is used for DNS requests, is enhanced by DoT with TLS encryption. Furthermore, the DoT protects against on-path attacks that alter or forge DNS requests and responses.

3.2.2 DNS-over-HTTPS (DoH)

DoH encrypts DNS queries and responses and mixes them with other HTTPS traffic on the same connection, which prevents malicious parties, as well as advertisers, ISPs, and others, from being able to interpret the data and makes DNS traffic analysis more difficult. Because DoH keeps user browsing, secure and private, attackers cannot forge or alter DNS traffic. DoH has been deployed by the most popular browsers. Additionally, clients can use these transports to send encrypted DNS queries to a third-party recursive resolver (e.g., Google or Cloudflare). DoH is an attractive protocol, providing improved confidentiality, integrity, and availability that the traditional DNS lacks.

DoH is proposed to mitigate the DNS privacy concerns and to protect the communication between end users and recursive resolvers. The DNS requests are encrypted using HTTPS. Similar to regular HTTPS, DoH also runs on TCP port 443. Browsers frequently offer DoH as an integrated module. Any applications that support HTTPS can send DoH queries because DoH uses the HTTPS protocol to transmit DNS requests and responses between stub resolvers and recursive resolvers. DoH queries are concealed within regular HTTPS traffic, making it difficult to block them without also blocking all other HTTPS traffic. Consequently, the hidden DNS requests within the larger flow of HTTPS traffic gives network administrators less visibility but provides users with more privacy and therefore effectively resists traffic analysis that only targets DNS.

3.2.3 DNS-over-QUIC (DoQ)

DNS over QUIC is referred to as DoQ, which is consistent with ”DNS Terminology” [39]. DoQ is a very promising protocol. Transport QUIC was proposed as an acronym for ”Quick UDP internet Connections”. In May 2022, DNS-over-QUIC was published as an RFC and assigned the number 9250. Since then, DNS-over-QUIC was officially treated as a proposed standard and has developed into a new Internet transport protocol that reduces latency and is dependable, secure, and planned to replace TCP (currently the most popular transport). Protocol negotiation, stream-based multiplexing, and flow control are provided by QUIC.

It has been determined that DNS-over-QUIC (DoQ) is sufficiently stable and has received enough positive community feedback for global use. Since QUIC is used by HTTP/3, the third major version of the Hypertext Transfer Protocol, a proposed solution has been proposed. DoQ and HTTP-over-QUIC were eventually renamed HTTP/3 and DoH3, respectively, while HTTP/2 has primarily been employed with TLS over TCP. The same semantics are supported by HTTP/3 over the new QUIC transport protocol.

It is a multiplexed transport protocol instead of TCP, as the previous iterations were. QUIC is faster, more dependable, and offers better encryption than TCP. Consequently, support for the DNS-over-HTTP/3 (DoH3) protocol on Android 11 and later versions was added by one of the top DNS service providers. DoH3 has several improvements; therefore, one of the major DNS service providers has added support for the DNS-over-HTTP/3 (DoH3) protocol on Android 11 and later versions. DoH3 has many improvements, including a solution for the ”head-of-line blocking” issue, which slows internet data transactions when a packet is lost or reordered, a frequent occurrence when using a mobile device and switching connections.

By combining the transport and encryption handshakes into a single round trip, QUIC reduces the connection establishment time, offers multiplexing, mandates encryption, addresses head-of-line blocking, and features mandatory encryption. QUIC was designed with DNS privacy and low latency. DoQ wants to replace all other DNS protocols that are currently in use because they lack privacy and/or necessitate more handshake round trips. DoQ outperforms DoT and DoH, making it the best choice for encrypted DNS.

3.2.4 DNS security extension (DNSSEC)

DNS Security Extensions (DNSSEC) was introduced in 1997. DNSSEC adds cryptographic authentication to validate the authenticity and integrity of an answer to a DNS query by providing data integrity to traditional DNS by adding a set of new DNS resource records [40]. These new features are listed as follows:

- RRSIG (resource record signature) records.
- DNSKEY records use two key pairs: the key signing key (KSK) and zone signing key (ZSK)
- DS (Delegation signer)
- NSEC: when a client makes a DNS query and either the name does not exist or if the resource record type requested does not exist, the NSEC record is returned as a negative answer.
- NSEC3: Resource Record which can be used as an alternative to NSEC and mitigates any issues might exist in NSEC [41].

4. AWARENESS FRAMEWORK

In this section, the proposed framework is presented in the form of pros and cons of each selected DNS threat prevention method. These pros and cons show a technical framework that cybersecurity professionals can use to accomplish their cybersecurity plans in terms of providing awareness for a secure DNS system. The experiment that has been conducted for evaluating the proposed framework aimed to investigate the effectiveness of DNS-over-Encryption (DoE) in preventing DNS threats. To do this, we conducted a series of experiments in which we simulated various types of DNS attacks and measured the efficiency of DoE in preventing them. Our results showed that DoE is able to effectively prevent DNS spoofing, cache poisoning, and man-in-the-middle attacks. In all of the simulated attacks, DoE was able to correctly identify and block the malicious traffic, while allowing legitimate traffic to pass through. In addition to its ability to prevent attacks, we also found that DoE had minimal impact on the overall performance of the DNS system. The time required to resolve a domain name was only slightly longer when using DoE, and the overall number of successful queries was not significantly affected.

The results suggest that DoE is a viable solution for preventing DNS threats. Not only does it effectively block malicious traffic, but it also has minimal impact on the overall performance of the DNS system, it found to be in testing had minimal impacts and in many cases DoH is faster, the time required to resolve a domain name was only slightly longer when using DoE, and the overall number of successful queries was not significantly affected.

Our results provide strong evidence for the effectiveness of DoE in preventing DNS threats. As such, it is our recommendation that organizations and individuals consider implementing DoE to protect their DNS infrastructure. In the following section, the framework is illustrated:

4.1 Analysis of pros and cons of DoT

Pros of DoT:

1. Improved security: DoT encrypts DNS traffic using TLS, preventing third parties from intercepting and modifying DNS queries and responses.

2. Improved reliability: DoT provides a security countermeasure against poisoning/spoofing and supports large payloads, hence more effectively mitigating hijacking and reflection distributed denial-of-service attacks [42].

3. DoT gains extensive support from advanced operating systems, such as Android 9+, from different DNS systems, Unbound and Stubby, and from large public DNS resolvers, like Cloudflare, Google public DNS and Quad9.

4. DoT offers two usage profiles if (Opportunistic Privacy profile) utilized, fallback options enable, hence the best DNS service, at the cost of no attack mitigation if encryptions are not applied [43].

Cons of DoT:

1. Performance overhead: Encrypting DNS traffic using TLS introduces extra query time overhead compared to DNS-over-UDP, which potentially impacts DNS performance.

2. Limited adoption: Similar to DNSSEC, DoT has experienced limited adoption among DNS resolvers and clients, which lead to interoperability issues.

3. Extra employment: DoT requires extra changes, including updating the OS or installing a specialized stub resolver and manual configuration of DoT resolvers.

4. Traffic analysis: DoT uses port 853 for communication. The use of a dedicated port could make DoT requests and/or responses distinguishable from other traffic, which gives network administrators less visibility but provides users with more privacy.

5. Downgrade attack: The client (e.g., agent) is forced to fall back to plaintext DNS, where an opportunistic privacy profile by default is enabled [44].

4.2 Analysis of pros and cons of DoH

Pros of DoH:

1. Improved security: DoH encrypts DNS queries and responses, preventing third parties from intercepting and modifying DNS traffic.

2. Improved reliability: DoH provides a security countermeasure against poisoning/spoofing and supports large payloads, hence more effectively mitigating hijacking and reflection distributed denial-of-service attacks [45].

3. Improved performance: DoH improves DNS performance by reducing the number of intermediaries involved in the DNS resolution process. The deployment overhead of DoH is much lower on the client side [46].

4. DoH has gained extensive support from the most popular browsers (e.g., Firefox, Google Chrome, and EDGE) and large public DNS resolvers (e.g., Cloudflare, Google public DNS and Quad9).

5. DoH offers one Strict-Privacy-profile without fallback options, hence the best attack mitigation, at the cost of DNS service if the fallback fails [47].

Cons of DoH:

1. Inherited limitations of the traditional TCP: latency at the start of data transmission in comparison to UDP, around-trip delay, and issues such as TCP's head-of-line blocking and missing multiplexing support on the transport layer.

2. Server authentication is required for optimal attack mitigation at the cost of no DNS service when fallback fails.

4.3 Analysis of pros and cons of DoQ

Pros of DoQ:

1. Improved security: DoQ encrypts DNS queries and responses, preventing third parties from intercepting and modifying DNS traffic.

2. Improved reliability: QUIC tries to mitigate amplification attacks by requiring that the initial packet must be at least 1200 bytes and that a server must not send more than three times the size of the request in response, in addition to all previous attacks.

3. Avoids HoL blocking in application and at the transport layer.

4. Improved performance: DoQ helps reduce DNS latency and improve performance.

Cons of DoQ:

1. Fallback to Other Protocols on Connection Failure: If the DoQ connection fails, clients attempt to fall back to the DoT and then potentially clear text if the Opportunistic Privacy profile is selected.

2. Limited adoption: DoQ is a relatively new protocol and has not yet experienced widespread adoption, which leads to interoperability issues with other DNS protocols.

3. Browser compatibility: DoQ requires support from both the client side and server side and may not be supported by all browsers.

4.4 Analysis of pros and cons of DNSSEC

Pros of DNSSEC:

1. Improved security: DNSSEC adds an extra layer of security to the DNS by using digital signatures to verify the authenticity of DNS records, which prevents attackers from altering DNS records and redirecting users to malicious sites.

2. Improved reliability: DNSSEC also helps prevent DNS cache poisoning attacks, where an attacker injects false DNS records into the DNS cache, which causes DNS queries to fail or be directed to incorrect destinations.

3. Mitigate Zone enumeration: NSEC3 White lies [48] prevent zone walking attacks by making it difficult for attackers to determine the names of all of the zones that are signed by a particular DNSSEC key. This is because NSEC3 White Lies server signs on-the-fly an NSEC record for the previous and next names in the zone, rather than the actual names of the surrounding zones.

Cons of DNSSEC:

1. Complex implementation: DNSSEC requires additional infrastructure and expertise to set up and maintain, which can be a barrier for some organizations.

2. Limited adoption: Despite its benefits, DNSSEC adoption is still relatively low, especially among smaller organizations, which leads to interoperability issues, as DNSSEC-secured domains may not be able to communicate with non-DNSSEC domains.

3. Denial-of-service attacks can still be performed, which will often suffice to satisfy a censor's objective. DOS occurs because the resolver will attempt to process the attacker's packet, determine that the DNSSEC signature is absent or invalid, and immediately return ”Bogus”.

4. NSEC introduces a sideeffect in that the contents of a zone can be enumerated which known as zone enumeration vulnerability. This property introduces undesired policy issues. Also, NSEC3 RRs are still susceptible to dictionary attacks [41].

5. Depriving the client from the ability to connect to the host corresponding to the name

5. KNOWLEDGE GRAPH AS AWARENESS TOOL

This section illustrates the explanation of developing a knowledge graph. Datasets of DNS threats from MITRE and NIST as an awareness tool has been presented. A public record of every CVE, CWE, CAPEC, and CPE provided by MITRE and NIST and loaded into Neo4j [49] using a Python script named GraphKer it is a free and open source tool, that provides a detailed and updated cybersecurity graph database using Neo4j. Algorithm 1 shows the Neo4j code for extracting data from datasets

ALGORITHM 1: NEO4J CODE FOR EXTRACTING DATA FROM DATASETS

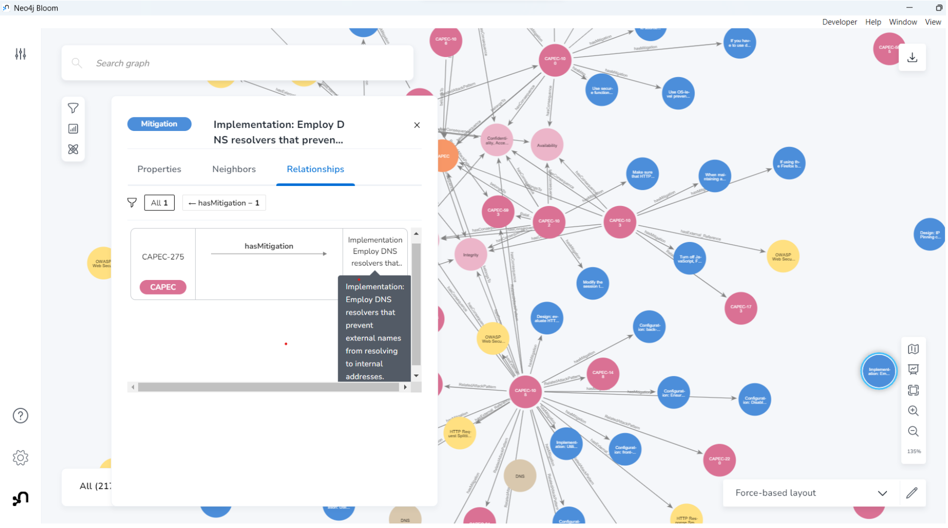
```
Connect to datasets
If connection
    Collect CVE, CWE, CAPEC, and CPE files
    Collect DNS threats
    Generate f
Else
    Send error
Open Neo4j
If connection
    Insert f to josn database
    Update user privileges
Else
    Send error
```

After all datasets are loaded into the database it can be examined using an exploration tool like Neo4j Bloom or querying it using Cypher. For the visual graph examples, we query the database with the Neo4j browser using the Cypher Query Language (CQL). Finally, we extract knowledge from connected graphs and limit the knowledge graph to the scope of our research domain (DNS threats). We produced the required datasets and those datasets [19] were equipped with the following properties: Publication Year, Reference ID, Author, Title, URL, Publisher, Version, Schema, Date, Name, Status, Resources Required, Description, Likelihood of Attack, Typical Severity, Submission Name, Extended Name, Skills Required Description, Skills Required, Abstraction, Submission Organization, Modifications, Prerequisites, Submission Date, Mitigations, Examples, Indicators, Notes, Alternate Terms, Scope. Fig. 2 shows Knowledge graph of DNS in Neo4j. Fig. 3 shows one of the relationships in DNS-KG. Fig. 4 shows Bitsquaitting DNS hijacking.

In the following, detailed descriptions of Fig. 2, 3, and 4 are provided. Fig. 2 represents the collection of public records from MITRE and NIST for CVE, CWE, CAPEC, and CPE, connected using Neo4j. We utilized GraphKer, a Python tool, to scrape the MITRE and NIST databases and load the data into Neo4j. Additionally, the data is regularly published as a Neo4j database backup file, which we load into Neo4j. This enables us to efficiently explore the data using tools like Neo4j Bloom or query it using cypher.

Fig. 3 demonstrates the linkage between the data in our network graph and the described data. It illustrates how we connect software running on our machines to the corresponding vulnerability database entries and automatically update the data. This enables us to efficiently identify the latest vulnerabilities and proactively apply relevant fixes.

Fig. 4 showcases the capability to delve into connections in-depth. For instance, starting with a bitsquatting threat, we can identify all the reported alerts available in CAPEC and then examine all the connected measurements to understand the potential scale of the bitsquatting attack.



Mitigation Implementation: Employ D NS resolvers that preven... ×

Properties Neighbors Relationships

All 1 ← hasMitigation – 1

CAPEC-275 CAPEC	hasMitigation →	Implementation Employ DNS resolvers that..

Implementation: Employ DNS resolvers that prevent external names from resolving to internal addresses.

Fig. 2. Knowledge graph of DNS in Neo4j.

5.1 Recommended usages of cybersecurity knowledge graphs

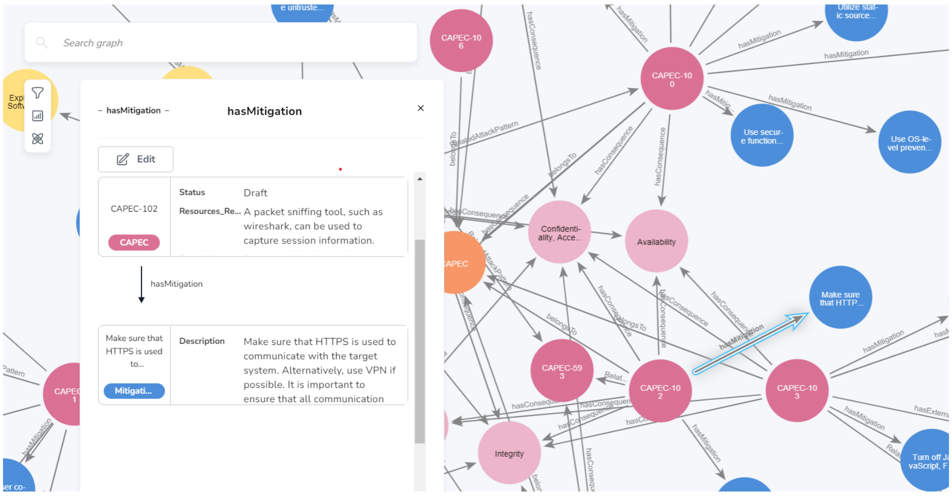
- Threat detection and response: Cybersecurity knowledge graphs can be used to detect and respond to threats by correlating data from different sources, such as vulnerability databases, threat intelligence feeds, and security logs. This can help to identify potential threats early on and take steps to mitigate them.

- Incident response: Cybersecurity knowledge graphs can be used to improve incident response by providing a centralized repository of information about incidents, such as the affected systems, the vulnerabilities exploited, and the steps taken to remediate the incident. This can help to speed up the response process and reduce the impact of incidents.

- Risk management: Cybersecurity knowledge graphs can be used to assess and manage risk by providing a comprehensive view of the organization's security posture. This can help to identify areas where the organization is at risk and take steps to mitigate those risks.

- Compliance: Cybersecurity knowledge graphs can be used to help organizations comply with security regulations by providing a way to track and manage compliance requirements. This can help to ensure that the organization is in compliance with regulations and avoid penalties.

- Security awareness and training: Cybersecurity knowledge graphs can be used to improve security awareness and training by providing a way to share information



CAPEC-102

Status	Draft
Resources_Re...	A packet sniffing tool, such as wireshark, can be used to capture session information.

Make sure that HTTPS is used to...

Description	Make sure that HTTPS is used to communicate with the target system. Alternatively, use VPN if possible. It is important to ensure that all communication

Fig. 3. One of the relationships in DNS-KG.

about security threats and best practices. This can help to educate employees about security risks and how to mitigate them.

6. DISCUSSION AND CONCLUSION

Complementary DNS over encryption (DoE) and DNSSEC are two mechanisms that effectively secure the domain name system (DNS). DoE ensures that DNS queries and responses are encrypted, preventing eavesdropping and interception by third parties. This approach helps protect users' privacy and prevent hackers from redirecting traffic to malicious sites. DNSSEC, on the other hand, adds a layer of security to the DNS by authenticating DNS records and responses, ensuring that they have not been tampered with or altered in transit. DNSSEC uses digital signatures and public key infrastructure (PKI) to verify the integrity of DNS data and to protect against spoofing attacks. Hence, collectively, DoE and DNSSEC provide a strong defense against DNS attacks and ensure the security and reliability of the DNS system.

In addition, our proposed framework provides recommendations to improve the privacy, security, and performance of DNS resolution from the client perspective:

- Use of a DNS-over-HTTPS (DoH) or DNS-over-TLS (DoT) service: These services encrypt DNS queries and responses, providing improved privacy and security. Popular DoH/DoT services include Cloudflare's 1.1.1.1 and Google's 8.8.8.8.

- Use of a secure DNS resolver: Secure DNS resolvers, such as Quad9 or OpenDNS, use a variety of security measures to protect against malicious DNS responses and other threats.

- Use of a local resolver: Local resolvers, such as Pi-hole or AdGuard, allow users

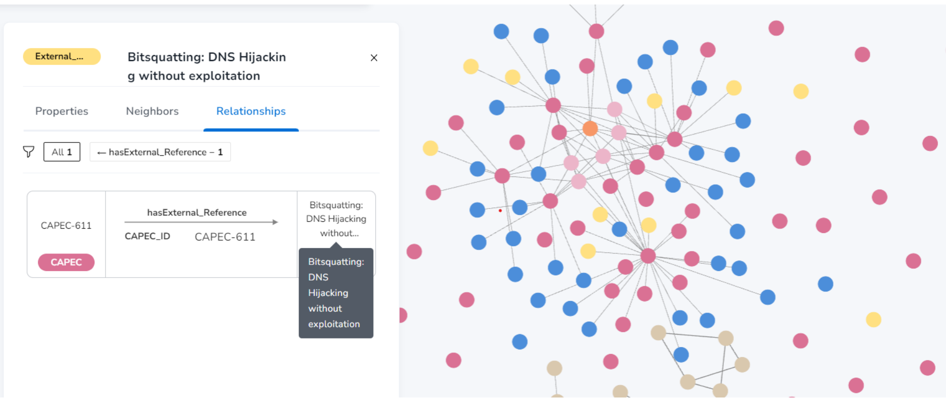

External_...
Bitsquatting: DNS Hijackin g without exploitation
×
Properties
Neighbors
Relationships
All 1
← hasExternal_Reference – 1
CAPEC-611
CAPEC
hasExternal_Reference
CAPEC_ID
CAPEC-611
Bitsquatting: DNS Hijacking without...
Bitsquatting: DNS Hijacking without exploitation

Fig. 4. Bitsquaitting DNS hijacking.

to block unwanted content and improve performance by locally caching DNS responses.

- Use of a virtual private network (VPN): VPNs encrypt all internet traffic and provide additional security and privacy by hiding the user's IP address and DNS queries.

Currently, the HTTPS protocol is well known. Motivations that triggered the revolution of secure DNS protocols have been initiated from a simple question: ”Does DNS over HTTPS satisfy the requirements?” The answer to the previous question is no, mainly because HTTP is not a transport layer protocol. HTTP serves as a substitute for a proper transport protocol, which would raise many risks, such as HTTP cookies, other HTTP headers (authentication, user-agent, and acceptance language), additional fingerprinting opportunities for malefactors, and tracking using ETag. Hence. There is a critical need to innovate a new practical DNS threat prevention method.

In this paper, a technical framework for preventing DNS threats has been suggested and presented in the form of usage guidance. The Experiments related to this paper are available on the GitHub [50]. Our proposed awareness framework is demonstrated by constructing a knowledge graph from well-known threat datasets such as CVE, CWE, CAPEC, and CPE, which are provided by MITRE and NIST. These datasets are widely recognized and contain a comprehensive range of discovered DNS threats, ensuring the generalizability of our proposed framework. The contributions of this work are to be summarized as follows:

1. Critical and analytical discussion for DNS security-related works is provided.

2. A comprehensive analysis of DNS threats, which provides a clear understanding of each DNS threat, is performed. This analysis could serve as a reference for DNS threats and assist professionals and academics in developing a clear understanding of these threats and their impact. This analytical discussion of the related research has highlighted the research gap in fighting against DNS threats.

3. A technical framework as usage guidance for security professionals to overcome DNS threats is provided.

4. Recommendations to improve the privacy, security, and performance of DNS resolution from the client perspective are provided.

5. In addition, we have developed a benchmark based on the basic security triad, which includes confidentiality, integrity, and availability. Table 4 shows the benchmark for comparing DNS threat prevention methods. Table 4 shows the benchmark evaluation.

Table 4. Benchmark evaluation.

Prevention method	Confidentiality	Integrity	Availability
DNSSEC	✗	✓	✗
DNS over TLS	✓	✓	✓
DNS over HTTPS	✓	✓	✓
DNS over QUIC/HTTP3	✓	✓	✓

This work is limited to providing a contribution on DNS security from an end-user perspective. While there is a lack of framework establishment by service providers and institutions to fully implement a comprehensive protection system involving all parties, end users are unable to apply the protection methods discussed in our study without the support of service providers and institutions. DNSSEC can be considered as an example.

The second limitation lies in manually connecting the knowledge graph tool (Neo4j) with DNS threats (CVE, CWE, CAPEC, and CPE). Therefore, there is a need to provide an automatic, dynamic solution that establishes connections with these mentioned databases and updates itself whenever new information becomes available in those datasets. The third limitation is that our experiments were conducted in a controlled environment, and further studies are needed to investigate the effectiveness of DoE in a real-world setting. Finally, while the present study focused on the most common types of DNS attacks, there may be other types of threats that DoE is unable to prevent.

REFERENCES

1. E. Nygren, R. K. Sitaraman, and J. Sun, “The akamai network: a platform for high-performance internet applications,” *ACM SIGOPS Operating Systems Review*, Vol. 44, no. 3, 2010, pp. 2–19.
2. S. Al-Mashhadi, M. Anbar, S. Karuppayah, and A. K. Al-Ani, “A review of botnet detection approaches based on dns traffic analysis,” *Intelligent and Interactive Computing: Proceedings of IIC 2018*, 2019, pp. 305–321.
3. G. D’Angelo, A. Castiglione, and F. Palmieri, “Dns tunnels detection via dns-images,” *Information Processing & Management*, Vol. 59, no. 3, 2022, p. 102930.
4. K. Fukuda, Y. Aharen, S. Sato, and T. Mitamura, “Characterizing dns query response sizes through active and passive measurements,” in *NOMS 2022-2022 IEEE/IFIP Network Operations and Management Symposium*. IEEE, 2022, pp. 1–6.

5. L. Mastilak, P. Helebrandt, M. Galinski, and I. Kotuliak, "Secure inter-domain routing based on blockchain: A comprehensive survey," *Sensors*, Vol. 22, no. 4, 2022, p. 1437.
6. A. Moubayed, M. Injadat, A. Shami, and H. Lutfiyya, "Dns typo-squatting domain detection: A data analytics & machine learning based approach," in *2018 IEEE Global Communications Conference (GLOBECOM).* IEEE, 2018, pp. 1–7.
7. R. Hananto, C. Lim, and H. P. Ipung, "Detecting network security threats using domain name system and netflow traffic," in *Proceedings of the 2nd International Conference on Cryptography, Security and Privacy*, 2018, pp. 105–109.
8. J. Spaulding, J. Park, J. Kim, and A. Mohaisen, "Proactive detection of algorithmically generated malicious domains," in *2018 International Conference on Information Networking (ICOIN).* IEEE, 2018, pp. 21–24.
9. S. Y. Chau, O. Chowdhury, V. Gonsalves, H. Ge, W. Yang, S. Fahmy, and N. Li, "Adaptive deterrence of dns cache poisoning," in *Security and Privacy in Communication Networks: 14th International Conference, SecureComm 2018, Singapore, Singapore, August 8-10, 2018, Proceedings, Part II.* Springer, 2018, pp. 171–191.
10. G. Mittal and V. Gupta, "Karmanet: Sdn solution to dns-based denial-of-service," in *Security in Computing and Communications: 6th International Symposium, SSCC 2018, Bangalore, India, September 19–22, 2018, Revised Selected Papers 6.* Springer, 2019, pp. 431–442.
11. A. Almusawi and H. Amintoosi, "Dns tunneling detection method based on multilabel support vector machine," *Security and Communication Networks*, Vol. 2018, 2018, pp. 1–9.
12. C. Deccio, A. Hilton, M. Briggs, T. Avery, and R. Richardson, "Behind closed doors: a network tale of spoofing, intrusion, and false dns security," in *Proceedings of the ACM Internet Measurement Conference*, 2020, pp. 65–77.
13. K. Bumanglag and H. Kettani, "On the impact of dns over https paradigm on cyber systems," in *2020 3rd International Conference on Information and Computer Technologies (ICICT).* IEEE, 2020, pp. 494–499.
14. A. Majumdar, S. Raj, and T. Subbulakshmi, "Arp poisoning detection and prevention using scapy," in *Journal of Physics: Conference Series*, Vol. 1911, no. 1. IOP Publishing, 2021, p. 012022.
15. K. Hynek, D. Vekshin, J. Luxemburk, T. Cejka, and A. Wasicek, "Summary of dns over https abuse," *IEEE Access*, Vol. 10, 2022, pp. 54 668–54 680.
16. A. Nadler, R. Bitton, O. Brodt, and A. Shabtai, "On the vulnerability of anti-malware solutions to dns attacks," *Computers & Security*, Vol. 116, 2022, p. 102687.
17. S. K. Singh and P. K. Roy, "Malicious traffic detection of dns over https using ensemble machine learning," *International Journal of Computing and Digital Systems*, Vol. 11, no. 1, 2022, pp. 189–197.
18. D.-A. Byamukama and J. Ngubiri, "Ddos amplification attacks and impacts on enterprise service-oriented network infrastructures: Dns servers," *International Journal of Information and Computer Security*, Vol. 18, no. 1-2, 2022, pp. 105–132.
19. Albluwi. Our dns datasets on the github. [Online]. Available: https://github.com/AbdalhadiB/DNS_KG
20. K. Man, Z. Qian, Z. Wang, X. Zheng, Y. Huang, and H. Duan, "Dns cache poisoning attack reloaded: Revolutions with side channels," in *Proceedings of the 2020 ACM*

SIGSAC Conference on Computer and Communications Security, 2020, pp. 1337–1350.

21. M. Dooley and T. Rooney, *DNS Security Management*. John Wiley & Sons, 2017.
22. M. Fejrskov, J. M. Pedersen, and E. Vasilomanolakis, "Detecting dns hijacking by using netflow data," in *2022 IEEE Conference on Communications and Network Security (CNS)*. IEEE, 2022, pp. 273–280.
23. D. Atkins and R. Austein, "Threat analysis of the domain name system (dns)," Technical Report, 2004.
24. M. Anagnostopoulos, G. Kambourakis, P. Kopanos, G. Louloudakis, and S. Gritzalis, "Dns amplification attack revisited," *Computers & Security*, Vol. 39, 2013, pp. 475–485.
25. R. Yazdani, R. van Rijswijk-Deij, M. Jonker, and A. Sperotto, "A matter of degree: characterizing the amplification power of open dns resolvers," in *International Conference on Passive and Active Network Measurement*. Springer, 2022, pp. 293–318.
26. K. Alieyan, A. Almomani, M. Anbar, M. Alauthman, R. Abdullah, and B. B. Gupta, "Dns rule-based schema to botnet detection," *Enterprise Information Systems*, Vol. 15, no. 4, 2021, pp. 545–564.
27. J. Park, M. Mohaisen, and A. Mohaisen, "Investigating dns manipulation by open dns resolvers," in *Proceedings of the 15th International Conference on emerging Networking EXperiments and Technologies*, 2019, pp. 45–46.
28. K. Schomp, T. Callahan, M. Rabinovich, and M. Allman, "Assessing dns vulnerability to record injection," in *International Conference on Passive and Active Network Measurement*. Springer, 2014, pp. 214–223.
29. D. Kaminsky, "Black ops 2008: It's the end of the cache as we know it," *Black Hat USA*, Vol. 2, 2008.
30. D. Dagon, M. Antonakakis, P. Vixie, T. Jinmei, and W. Lee, "Increased dns forgery resistance through 0x20-bit encoding: security via leet queries," in *Proceedings of the 15th ACM conference on Computer and communications security*, 2008, pp. 211–222.
31. M. Kührer, T. Hupperich, J. Bushart, C. Rossow, and T. Holz, "Going wild: Large-scale classification of open dns resolvers," in *Proceedings of the 2015 Internet Measurement Conference*, 2015, pp. 355–368.
32. W. Xu, K. Sanders, and Y. Zhang, "We know it before you do: Predicting malicious domains," 2014. [Online]. Available: https://api.semanticscholar.org/CorpusID:114611131
33. T. Holz, C. Gorecki, K. Rieck, and F. C. Freiling, "Measuring and detecting fast-flux service networks." in *Ndss*, 2008.
34. S. Schiavoni, F. Maggi, L. Cavallaro, and S. Zanero, "Phoenix: Dga-based botnet tracking and intelligence," in *International Conference on detection of intrusions and malware, and vulnerability assessment*. Springer, 2014, pp. 192–211.
35. P. Agten, W. Joosen, F. Piessens, and N. Nikiforakis, "Seven months' worth of mistakes: A longitudinal study of typosquatting abuse," in *Proceedings of the 22nd Network and Distributed System Security Symposium (NDSS 2015)*. Internet Society, 2015.

36. N. Nikiforakis, S. Van Acker, W. Meert, L. Desmet, F. Piessens, and W. Joosen, "Bitsquatting: Exploiting bit-flips for fun, or profit?" in *Proceedings of the 22nd international conference on World Wide Web*, 2013, pp. 989–998.
37. P. Kintis, N. Miramirkhani, C. Lever, Y. Chen, R. Romero-Gómez, N. Pitropakis, N. Nikiforakis, and M. Antonakakis, "Hiding in plain sight: A longitudinal study of combosquatting abuse," in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, 2017, pp. 569–586.
38. A. Khormali, J. Park, H. Alasmary, A. Anwar, M. Saad, and D. Mohaisen, "Domain name system security and privacy: A contemporary survey," *Computer Networks*, Vol. 185, 2021, p. 107699.
39. M. Lyu, H. H. Gharakheili, and V. Sivaraman, "A survey on dns encryption: Current development, malware misuse, and inference techniques," *ACM Computing Surveys*, Vol. 55, no. 8, 2022, pp. 1–28.
40. N. K. Nainar and A. Panda, "Capturing secured application traffic for analysis," *Wireshark for Network Forensics: An Essential Guide for IT and Cloud Professionals*. Springer, 2022, pp. 65–105.
41. R. Arends, G. Sisson, D. Blacka, and B. Laurie, "Dns security (dnssec) hashed authenticated denial of existence," *DNS Security (DNSSEC) Hashed Authenticated Denial of Existence*, 2008.
42. M. Polese, F. Chiariotti, E. Bonetto, F. Rigotto, A. Zanella, and M. Zorzi, "A survey on recent advances in transport layer protocols," *IEEE Communications Surveys & Tutorials*, Vol. 21, no. 4, 2019, pp. 3584–3608.
43. D. Sharma and A. K. Tyagi, "Preserving privacy in internet of things (iot)-based devices," in *Proceedings of Third International Conference on Computing, Communications, and Cyber-Security: IC4S 2021*. Springer, 2022, pp. 803–816.
44. S. Dickinson, D. Gillmor, and T. Reddy, "Usage profiles for dns over tls and dns over dtls," Technical Report, 2018.
45. Q. Huang, D. Chang, and Z. Li, "A comprehensive study of {DNS-over-HTTPS} downgrade attack," in *10th USENIX Workshop on Free and Open Communications on the Internet (FOCI 20)*, 2020.
46. T. Wirtgen, T. Rousseaux, Q. De Coninck, N. Rybowski, R. Bush, L. Vanbever, A. Legay, and O. Bonaventure, "{xBGP}: Faster innovation in routing protocols," in *20th USENIX Symposium on Networked Systems Design and Implementation (NSDI 23)*, 2023, pp. 575–592.
47. N. Nikiforakis, M. Balduzzi, L. Desmet, F. Piessens, and W. Joosen, "Soundsquatting: Uncovering the use of homophones in domain squatting," in *Information Security: 17th International Conference, ISC 2014, Hong Kong, China, October 12-14, 2014. Proceedings 17*. Springer, 2014, pp. 291–308.
48. D. Kaminsky. Dan kaminsky's. [Online]. Available: https://dankaminsky.com/phreebird/
49. Neo4j. Graph databases data science. [Online]. Available: https://neo4j.com/product/graph-data-science/
50. Albluwi. Our experiment on the github. [Online]. Available: https://github.com/AbdalhadiB/Experiment2.1

PLACE
PHOTO
HERE

Abdulhadi AlBluwi received his BS in Information Technology from the Saudi Electronic University, Tabuk, KSA, in 2018 and his MS degree in Information Security from University of Tabuk, Tabuk, KSA in 2023. His research interests include security in network, DNS and database, and machine learning.

PLACE
PHOTO
HERE

Umar Albalawi received his Ph.D. degree in Computer Science and Engineering from the University of North Texas in 2016, and master's degree in Computer Science from Texas AM University, in 2013. He is currently an Associate Professor in the Department of Information Technology, University of Tabuk, Saudi Arabia. His research interests focus on Security and Privacy in Internet of Things (IoT), Network Security, and Cryptography. He served on the Editorial Boards of Several peer-reviewed international journals and magazine.

PLACE
PHOTO
HERE

Abdelrahman Osman Elfaki is currently associate professor at Information Technology Department, University of Tabuk, KSA. Before, he has been a senior lecturer at Management and Science University (MSU) in Malaysia. He was involved in many IT projects and research in different countries which has refined his experience in both practical and academic fields. He has published many papers in high reputable journals and conferences.