Abstract

The domain name system (DNS) is one of the most important components of today's Internet, and is the standard naming convention between human-readable domain names and machine-routable IP addresses of Internet resources. However, due to the vulnerability of DNS to various threats, its security and functionality have been continuously challenged over the course of time. Although, researchers have addressed various aspects of the DNS in the literature, there are still many challenges yet to be addressed. In order to comprehensively understand the root causes of the vulnerabilities of DNS, it is mandatory to review the various activities in the research community on DNS landscape. To this end, this paper surveys more than 170 peer reviewed papers, which are published in both top conferences and journals in last ten years, and summarizes vulnerabilities in DNS and corresponding countermeasures. This paper not only focuses on the DNS threat landscape and existing challenges, but also discusses the utilized data analysis methods, which are frequently used to address DNS threat vulnerabilities. Furthermore, we looked into the DNS threat landscape from the view point of the involved entities in the DNS infrastructure in an attempt to point out more vulnerable entities in the system.

1. Introduction

The Domain Name System (DNS) is one of the pillars of the operation of the Internet, the medium on which most of communications are today transported. Among other purposes, DNS is used today widely for domain names to translate them into Internet Protocol (IP) addresses. The domain name system maps a name that people use to locate a website to the IP address that a computer uses to locate that website. For example, if a user types www.example.com into a web browser, a server behind the scenes will map that name to an IP address, e.g., 1.2.3.4, corresponding to that domain. Web browsing and most other Internet activities, e.g., transferring files, rely on DNS to quickly provide the information necessary to connect users to remote hosts. DNS mapping is distributed throughout the Internet in a hierarchy of authorities.

Based on the key role of the DNS in networking infrastructure, attackers are aggressively looking for new ways to compromise the DNS infrastructure. Therefore, it is necessary to understand the evolution of DNS security and the associated issues to preserve reliable and secure services, and to improve the security of DNS through various iterations of design revisions. Indeed, recently, researchers from academia and industry have focused on improving the design of DNS by allowing various necessary options, and by revising its assumptions of operation. However, the constant evolution of DNS, as well as the rise of new detrimental and unconventional issues, such as pervasive adversaries, privacy risks, and new and advanced forms of attacks, make the functionality, security, and privacy of DNS important issues that require continued attention and investigation. To this end, surveying, summarizing, and categorizing the body of work on DNS security and privacy is an important endeavor that is lacking in the existing literature. This effort is necessary to guide the community to what is an open problem that requires further attention.

Particularly, despite the large number of works in the literature on DNS security and privacy, existing contributions are scattered across different research areas and a comprehensive yet concise survey is lacking. Although there exist several surveys in the field of DNS [1, 2, 3, 4, 5], they are limited from several points of view. First, the scope of those works is narrow, e.g., covering only a certain aspect of the DNS ecosystem, such as detecting Internet abuse by analyzing passive DNS traffic [5], malicious domain detection through DNS data analysis [4], and investigating the domain name squatting ecosystem [5]. Second, the number of studies covered in each of those prior works—even on the very well accepted set of problems in the field of DNS security and privacy—is limited, and does not cover the most recent advances, which require further attention and consideration—especially in an evolving domain [1, 5].

To this end, this survey fills the gap in the literature by putting together an up-to-date summary of research works that cover various aspects of the DNS ecosystem and associated security and privacy issues—and all in one place. Toward this goal, we have surveyed a wide range of research works from various points of view, including the DNS threat landscape, DNS research methods, and DNS research scopes.

The general outline of the survey, highlighting the road map towards the key terms and directions, is shown in Figure 1. In the following we briefly review some of those broad directions.

1.1. DNS Threat Landscape

A central component of this work is DNS threat landscape. By reviewing the threats, we primarily focus on the common
challenges in security and privacy of DNS infrastructure. Such threats may not only compromise the security and privacy of end-users, but may also undermine the performance and availability of Internet services that heavily rely on DNS for their operation. We demonstrate such an interdependence through various practical and challenging attacks and efforts in the literature to defend against them. For example, attackers abuse the functionality of open DNS resolvers to transfer a small number of DNS queries into significantly large payload coordinated at a target server or network, thus making the server inaccessible, in what is known as amplification attacks. Despite the large body of research works on detection and mitigation of DNS amplification attacks [6, 7, 8, 9, 10, 11, 12, 13, 14], such attacks still compromise today’s Internet, according to Nexusguard’s 2018 threat report of Q1 [15].

Another example we highlight is cache poisoning. DNS resolver cache can be compromised through the insertion of unauthorized domain names and IP addresses in order to redirect users to malicious IP addresses. Consequently, by launching this attack an adversary can compromise the performance of Internet services and distribute malicious contents, such as malware, spam, credentials theft, phishing contents, among many others. There exist several research works that attempted to address DNS cache poisoning attacks [16, 17, 18, 19, 20, 21, 22] using different methods, such as DNSSEC [23, 24], WSEC DNS [19], ECO-DNS [18], and Anax [17], among others, which we outline in this study.

Attackers also utilize botnets that allow them to access devices through propagation and coordinate among them using C&C channels to perform wide variety of malicious activities, such as Distributed Denial of Service (DDoS) attack [24], amplification attacks, and credentials stealing, among many others [25]. DNS plays an essential role in the operation of botnets by providing easily deployable C&C channel. Although various research works have studied botnets and possible solutions to defend against their attacks [26, 27, 28, 29], including marginalizing DNS infrastructure used for this attack. However, such attacks are still on the rise, compromising users’ security. Specifically, the emergence and rise of IoT and associated vulnerabilities opened new directions, whereby attack vectors facilitated by DNS’ reliability and deployment played a crucial role in those attacks. For example, the Mirai botnet attacked IoT devices, such as routers and video recorders [30, 31], and uses them for amplification attacks, necessitating security solutions optimized for IoT and its interplay with DNS [32, 33].

DNS manipulation may expose users to multiple threats, such as phishing and malicious domains. Accordingly, a large number of studies considered DNS manipulation [34, 35, 36, 37, 38], and its interplay with the detection and mitigation of phishing attacks [39, 40, 41, 42, 43, 44]. Despite this large number of studies, phishing and associated DNS manipulation is still a challenging and expanding area. Phishers are targeting mobile messaging and social apps at an increasing rate [45], highlighting the need for further investigation and defenses.

A large body of research has been devoted to the detection of malicious domain names [46, 47, 39, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58], which can be classified into two general categories: classification based [46, 47, 39, 48, 26] and inference based [54, 51, 55, 58]. Classification-based methods rely on local network and host information. The inference-based methods exploit global information along with local information of domains.

Domain Generation Algorithms (DGAs) are utilized to intermittently produce a large number of domain names for malicious purposes. There exist several research works that explored the ecosystem of DGAs [59, 60, 61, 62, 63].

Furthermore, another type of threat is domain name squatting, where cybersquatters register variants of popular trademark names through different squatting strategies, has been considered. Such attacks include various variants, i.e., typosquatting [63, 64, 5, 65, 66, 67, 68, 69], bitsquatting [70, 71, 72], combosquatting [41], and soundsquatting [73]. Domain name squatting becomes popular as an attack vector to start many other types of attacks [66, 65]. Related to this is domain monetization through parking; despite technically being a legitimate business, parked domain monetization can often lead to and get mixed with other practices, i.e., click fraud, as well as traffic spam and traffic stealing—which usually lead to serving malicious content i.e. malware and spam [74, 66, 75, 76]. All of these threats may compromise the security and privacy of DNS landscape and users. There are large amount of research regarding privacy leakage and various solutions have been proposed.

Figure 1: The general outline of the survey. We have surveyed a wide variety of research works from various points of view, including DNS threat landscape, DNS research methods, and DNS research scopes.
such as EncDNS [77], PPDNS [78], DLV [29], etc. However, DNS Research Methods. Another vertical of our survey is the DNS research methods used in the literature, and their associated work. In exploring the DNS research methods, we have investigated various data analysis methods, which have been utilized to detect, model, and mitigate the aforementioned DNS threats. First, we describe two main DNS data collection methods utilized in the literature and the associated works, including passive DNS data (PDNS) [80, 47, 81, 82, 83, 12, 11, 84, 48, 44, 67, 22, 51] and Active DNS data (ADNS) [67, 83, 85, 54, 41, 86]. Next, we categorize the research works based on their scope of analysis and from security point of view into vanilla DNS [85, 40, 29, 92, 93, 94, 18] and DNSSEC [95, 96, 97, 86, 98, 99, 23, 3, 100].

1.2. DNS Entities Scopes
We categorize the literature based on the entities they target, including DNS name servers, open DNS resolvers, hosting providers, and clients. Such categorization would reveal the most susceptible entity and its impact on the performance and security of whole DNS ecosystem. For example, open DNS resolvers can be exploited to conduct multiple attacks on behalf of attackers [101, 17, 46, 53, 22, 54, 55, 56, 87, 88, 89, 76] and association analysis [59, 82, 90, 54, 21]. Finally, we categorize the research works based on their scope of analysis and from security point of view into vanilla DNS [85, 40, 29, 92, 93, 94, 18] and DNSSEC [95, 96, 97, 86, 98, 99, 23, 3, 100].

1.3. Organization
In section 2 we present an overview of DNS functionality and DNS resolution. Major DNS threats and associated countermeasures are presented in section 3. DNS research methods are presented in section 4. In section 5 we explore research work addressing the different entities in the DNS model. Finally, the paper is concluded in section 6.

2. Preliminaries
This section is devoted to provide an overview of how DNS works, including the domain name resolution process.

2.1. Functionality
When a user wants to visit a website through the Internet he does so by connecting to a domain name, such as www.example.com; however, computers do not communicate with domain names, but rather they are only able to communicate using addresses represented as numbers, namely IP addresses. Therefore, the domain name of the intended website should be converted to its associated IP address. Addresses in the network can be represented either using IPv4 or IPv6. The first is composed of four bytes and is represented using four number segments, such as 1.2.3.4, whereas IPv6 is four times larger and is composed of eight segments, each of them is two bytes and represented by hexadecimal numbers. In comparison to natural languages, recognizing and remembering numerical addresses of the intended Internet services is not an easy task for users. Therefore, in the early 1980’s Paul Mockapetris [102] introduced the basics of the DNS, which enables users to automatically map human-readable domain names into machine-readable addresses. DNS is an essential component of the functionality of the Internet.

2.2. DNS Resolution
DNS is a hierarchical and globally distributed directory service that utilizes root, TLD, and authoritative name servers in order to resolve string addresses of the domains into IP addresses. The process of domain name resolution is shown in Figure 2. The DNS resolution process begins once a user attempts to access a web service using Internet web browser. In the case where there is no information about the given domain in the local cache and host table, the local resolver initiates a DNS query to a recursive resolver seeking to match the domain with its corresponding IP address. Then, the recursive resolver starts asking the root, TLD, and then authoritative name servers, and over steps ① through ⑦, to answer the query. The root server is the first server that receives queries from recursive resolvers, in step ①. The root servers are globally distributed and maintain the IP addresses and location of authoritative TLD name servers. In step ①, the root name server replies to the query with the appropriate list of authoritative TLD servers for the .com TLD. In step ②, the recursive server sends a query for example.com to the .com TLD name server. Once the query reaches the .com TLD name server, it responds with the IP address of the domain’s authoritative name server, as shown in step ①. At the next step, namely step ②, the recursive resolver sends a query to the authoritative name server. The authoritative name server knows the IP address for www.example.com and at step ③ that answer is returned to the recursive name server. Finally, the determined IP address of the requested domain name is forwarded to the local resolver and then to a web browser. As a result, the browser can send a Hypertext Transfer Protocol (HTTP) request to the website to retrieve its contents.

3. DNS Threat Landscape
The original implementation of the DNS did not include security and privacy protections; instead the original DNS was built to be a scalable distributed system. As the Internet has
Table 1: List of abbreviations in alphabetical order.

| Term       | Definition                      |
|------------|---------------------------------|
| 2LD        | Second Level Domain             |
| 3LD        | Third Level Domain              |
| AP         | Affinity Propagation            |
| BGP        | Border Gateway Protocol         |
| C&C        | Command and Control             |
| CPM        | Convex Polytope Machine         |
| DDoS       | Distributed Denial of Service   |
| DGA        | Domain Generation Algorithm     |
| DNS        | Domain Name System              |
| DNSSEC     | DNS Security Extensions         |
| DoS        | Denial of Service               |
| DT         | Decision Tree                   |
| HTTP       | Hypertext Transfer Protocol     |
| IoT        | Internet of Things              |
| IPv4       | Internet Protocol Version 4     |
| IPv6       | Internet Protocol Version 6     |
| ISP        | Internet Service Provider       |
| KNN        | K-Nearest Neighbour             |
| LAN        | Local Area Network              |
| LWL        | Locally Weighted Learning       |
| MITM       | Man in the Middle               |
| ML         | Machine Learning                |
| MLP        | Multi Layer Perceptron          |
| MS         | Mean Shift                      |
| NAT        | Network Address Translation     |
| NB         | Naive Bayes                     |
| NN         | Neural Networks                 |
| RBC        | Rule-Based Classifier           |
| RDNS       | Recursive Domain Name System    |
| RF         | Random Forest                   |
| RR         | Resource Records                |
| RSA        | Rivest Shamir Adleman           |
| SDN        | Software Defined Networking     |
| SL         | Simple Logistic                 |
| SVM        | Support Vector Machine          |
| TCP        | Transmission Control Protocol   |
| TLD        | Top Level Domain                |
| TLS        | Transport Layer Security        |
| TSV        | Transductive Support Vector Machine |
| TTL        | Time To Live                    |
| UDP        | User Datagram Protocol          |
| URL        | Uniform Resource Locator        |

Figure 2: Illustration of DNS resolution over recursive, root, TLD, and authoritative name server. A user attempts to access www.example.com (step 1), the client sends a query to the recursive resolver (step 2). The recursive resolver sends a query to the TLD server (step 3), which returns a response (step 4). The authoritative name server then returns a response (step 5). The client then updates its cache (step 6) and returns the result (step 7).

grown, however, adversaries have found weaknesses that can be abused for malicious purposes. There is a large body of research in this area, which lead to identification of major threats that substantially compromise the DNS security. In this section, we would delve into the common threats in the DNS ecosystem and state-of-the-art countermeasures presented in the literature.

Our effort in enumerating and categorizing the literature in this domain is featured in the follow. We review the proposed methods for detection and mitigation of popular threats of DNS infrastructure, including amplification and DoS attacks (§3.1), DNS cache poisoning attacks (§3.2), botnets attacks (§3.3), phishing attacks (§3.4), DNS manipulation (§3.5), malicious domain detection (§3.6), domain generation algorithms (§3.7), domain name squatting (§3.8), parked domain monetization and associated security issues (§3.9), and privacy leakage (§3.10).

In addition, we will discuss challenges and highlight open problems that require further investigations by security researchers associated with each of those major problems. In Table 9 and Table 10, topical classification of DNS threats in the literature is addressed, with sample works presented as well.

### 3.1. Amplification and DoS Attacks

A DNS amplification attack is a reflection-based DDoS attack. In DNS, an amplification attack is done by issuing a small number of DNS queries that are later transformed into a considerably large payload coordinated at the target network. The high level architecture of a typical DNS amplification attack is demonstrated in Figure 3. The attacker hides the exploit source and directs the DNS response into the target address through spoofing look-up requests issued to DNS servers. It is difficult to defend against such an attack, since it originates from legitimate servers with legitimate traffic.

A wide variety research works have been conducted to detect and mitigate DNS amplification attacks [7, 9, 103, 104, 10, 13, 6, 11, 12, 8, 14]. For instance, Ballani et al. [7] have presented a simple method based on caching behavior analysis of DNS resolvers to defend against DNS DoS attacks. They have stored cached records with TTL in a stale cache. Then, the stale cache can be used by a resolver that does not receive.
any response from the authoritative name servers. Herzberg et al. [9] have designed an anti-reflection system, providing DNS authentication, which nullifies the amplification factor of the DNS responses abused for DoS attacks. DNS authentication is composed of two subsystems, namely a request authentication that detects and filters requests sent from spoofed IP addresses, and resolver authentication that maintains a list of potentially compromised hosts. They have deployed the resolver authentication as a cloud-based service to further reduce costs and provide additional defenses for DNS servers. Furthermore, Rijswijk-Deij et al. [104] investigated the potential for abuse in DNSSEC-signed domains in a large scale, covering 70% of all signed domains in operation. Their analysis demonstrate that DNSSEC in fact empowers DNS amplification attacks for a particular query type, ANY. In addition, Rossow [103] has investigated Distributed Reflective Denial-of-Service (DRDoS) attacks through revisiting well-known UDP-based protocols, including network services, online games, file sharing networks, and botnets to assess their security against DRDoS abuse. His analysis revealed that attackers already started abusing 14 protocols through bandwidth amplification and multiplying the traffic up to a factor 4670.

MacFarland et al. [10] examined a large number of domains (≈129M) and authoritative servers (≈1.1M) to investigate the inherent DNS amplification risks associated with DNS authoritative name servers. Their analysis showed that only a small number of authoritative servers (≈3.8%) are responsible for the highest amplification factors. In addition, their analysis revealed that adoption of DNS response rate limit is limited to less than ≈3% of authoritative servers. Finally, they have suggested tunnelling into a remote resolver as a straightforward and simple countermeasure to mitigate on-going attack at the organization level. Verma et al. [13] have utilized the fact that DNS resolvers share the local DNS query rates to propose an amplified DNS attack mitigation system called Distributed Rate Sharing-based Amplified DNS-DDoS Attack Mitigation (DRS-ADAM). The authors claim that DRS-ADAM detects and completely stops an amplified DNS attack by imposing DNS query rate sharing among resolvers involved in an attack. DRS-ADAM has several advantages: deployment, robustness against manipulation, and attack mitigation.

Aizuddin et al. [6] have incorporated sFlow with security-centric SDN features to analyze DNS query identifiers (IDs) for detecting and mitigating DNS amplification attacks in a timely manner. Their analysis showed that the proposed method provides accurate detection results (more than 97.0%) even with a small number of flow values (DNS attributes). Kim et al. [8] have presented a DNS amplification attack mitigation system through a one-to-one strict mapping method between DNS requests and responses in order to identify orphan DNS responses. Their analysis showed that the proposed solution removes the possibility of false positive packets. Zheng et al. [14] have proposed Reinforcing Anti-DDoS Actions in Real-time (RADAR) which detects various DDoS attacks, such as link flooding, SYN flooding, and UDP-based amplification attacks. They do so through adaptive correlation analysis on commercial off-the-shelf SDN switches. RADAR does not require any changes in the SDN protocols and switches deployed in the network today, nor does it require additional appliances to detect attacks, making it an easy-to-plug solution in today’s operations.

Truong et al. [12] analyzed DNS traffic to design a detection system for recognition of pseudo-random domain names, including Conficker and Zeus, form legitimate domain names. The proposed detection system is composed of two main subsystems, including feature extraction and classification. The length of domain names and their expected values construct the feature space and classification section is composed of several classification algorithms, e.g., RF, KNN, SVM, and NB.

**Discussion.** Despite the large body of research work on the detection and mitigation of DNS amplification and DoS attacks, such attacks are still prevalent and compromising today’s Internet. Table 2 summarizes the proposed methods in the literature and their strengths and weaknesses. Shortcomings of the pro-

Table 2: Summary of representative work that studied DNS amplification attacks; RL is Response Latency, ID is Input Data, and O(1,2,3) donates Other disadvantages: (1) modification of standard DNS resolver semantics, (2) third-party organizations, and (3) low detection accuracy.

| Work | Method | Advantages | Disadvantages |
|------|--------|------------|---------------|
| [6]  | sFlow/SDN | Near-real-time detection | RL, ID, O(1,2) |
| [12] | Stale cache | Simple, lightweight | X |
| [12] | One-to-one mapping | Complete mitigation | X |
| [10] | DNS authentication | Cloud-based service | X |
| [9]  | Tunnel/remote resolver | Legacy compatible | X, O(2) |
| [10] | ML/domain features | Small feature space | X, O(2) |
| [13] | DRS-ADAM | Easy deployment | X |
| [13] | RADAR | Real-time detection | X |
posed methods, which would require further attention from the community, can be summarized as: 1. increasing response latency due to detection process, which requires light-weight and latency sensitive detectors [9, 10, 6, 8]. 2. requiring a range of changes to the DNS resolvers and semantics [17], which calls for work that address legacy-compatibility or require very little of such changes. 3. requiring large number of IP addresses to be collected every day [11], or requiring large number of flow rules [14], which calls for aggregate and light-weight feature engineering methods. 4. low detection accuracy, in some cases [12], which calls for improving accuracy through multimodality of detection features.

3.2. DNS Cache Poisoning

The DNS resolver cache, or simply DNS cache, is a temporary database that stores resolved DNS look-ups. As a result, this caching enables users to quickly resolve a previously visited website. Unfortunately, DNS cache can be compromised through the insertion of unauthorized domain names and IP addresses. General flow of the DNS cache poisoning attack is illustrated in Figure 4. In this manner, users’ queries might be directed to a fake destination with malicious content or advertisements. DNS cache poisoning compromises the correct operation of Internet services and can be used for malicious activities, such as distributing malware and spam, phishing attacks, credential theft, etc. To this end, several research works have been conducted to detect and mitigate the DNS cache poisoning attacks [16, 17, 19, 20, 18, 21, 22].

Herzberg and Shulman [23] studied the security of the patched DNS and found that not only source ports may be circumvented by various NAT devices, but also IP address randomization of standard-conforming resolvers can be circumvented. In addition, they demonstrated that DNS query randomization with both random prefix and 0x20 encoding can be circumvented easily. Finally, they suggested the deployment of DNSSEC as a countermeasure to prevent DNS cache poisoning attacks. Furthermore, Weaver et al. [34] have analyzed a large number of measurement sessions from distinct IP addresses, collected by NetaHyzer [105], to understand the DNS behavior. Their analysis revealed that DNS infrastructure suffers from significant limitations, such as inefficient look-ups, unreliability of IP-level fragmentation, and ISP-driven manipulation of DNS.

Shulman and Waidner [16] studied the security of DNS infrastructure and highlighted that although adoption of challenge-response defenses [106, 107] are prevalent, DNS infrastructure is highly exposed to cache poisoning attacks. The authors claimed that DNSSEC is a suitable solution to defend against cache poisoning attacks. Antonakakis et al. [17] have studied open recursive DNS resolvers’ cache poisoning attacks and have found that attackers generally point victims to rogue IP addresses. Therefore, they have proposed Anax, a system that examines the nature of cache poisoning attacks and automatically detects them. Anax analyses resource records and extracts a set of statistical features, such as domain name diversity, 2LD diversity, 3LD diversity etc. Then, these features are fed into a set of learning algorithms, such as SVM, neural network, etc. for cache poisoning attack detection.

Perdisci et al. [19] proposed Wildcard SECure (WSEC) DNS as a solution that defends against DNS cache poisoning attacks. WSEC DNS incorporates wildcard domains with TXT resource records to secure recursive DNS servers from cache poisoning attacks. WSEC DNS can be easily deployed, since it does not require any modifications of the root and TLD name servers.

Wu et al. [20] have proposed a DNS cache poisoning attack detection system based on the Kalman filter technique. The proposed system is composed of two subsystems: an entropy sequence, which is modeled using state space equations, and a second Kalman filter method, which is used to detect the attacks. The effectiveness of the proposed system is illustrated for both single and distributed cache poisoning attacks. The measurement errors and the correlation variation of the prediction errors are utilized for detection of the single and the distributed cache poisoning attacks, respectively.

Chen et al. [21] have investigated the impact of disposable domains on the caching behavior of the DNS resolvers. Their analysis showed that as the prevalence of the disposable domains, it is likely that the DNS cache begins to be filled with resource records that are unlikely to be reused. In addition, their analysis revealed that disposable domains not only are widely used by different industries, such as Anti-Virus

Table 3: Summary of the some of representative research works that have investigated DNS cache poisoning attacks. Here LDR is Low Deployment Rate of DNSSEC, AR is Accuracy Rate, and WSR donates

| Work            | Method       | Advantages                        | Disadvantages |
|-----------------|--------------|-----------------------------------|---------------|
| [16]            | DNSSEC       | A-posteriori detection            | LDR x         |
| [17]            | Anax         | Easy deployment                   | AR x          |
| [19]            | WSEC DNS     | Easy deployment                   | WSR x         |
| [23]            | DNSSEC       | Cryptographic/verifiable         |               |

Figure 4: DNS cache poisoning attack. Attacker injects fake DNS entries into the DNS cache resolvers in an attempt to direct users into fake destinations, mainly for malicious purposes like phishing.

1xLD refers to the x-th level domain. In example.com IP:4.3.2.1 example.com is a 2nd level domain, and www.example.com is a 3rd level domain.
companies, e.g., McAfee, Sophos, and popular websites, e.g., Microsoft, Google, social networks, e.g., Facebook, etc., but also are increasing in that trend.

Similarly, Hao and Wang [22] have studied the negative impact of one-time-use domain names on the performance of DNS caching. Therefore, they proposed a one-time-use domain detection system that incorporates syntactical features extracted from domain name string, such as length of query name, length of the longest sub-domain name, Sub-domain depth, etc. with machine learning algorithms is proposed. The authors argue that removal or even not inserting such resource records into the cache can prevent from waste of the DNS cache resources.

Discussion. Different research works have proposed various methods for detection and mitigation of DNS cache poisoning attacks. These methods and their strengths and weaknesses are summarized in Table 4. The limitations of proposed solutions can be summarized as: 1. low deployment rate of DNSSEC [23, 10], which by design address cache poisoning, calling for further exploration of how to creates incentives for the spread of DNSSEC deployment. 2. low detection accuracy [17, 22], calling for better accuracy using multiple types of features. 3. causing overheads on DNS traffic, memory usage and increasing response latency [19], calling for lightweight and more efficient approaches. 4. and wasting DNS cache storage resources by prefetching unpopular resource records [13], which calls for adaptive resolution.

Table 4: Summary of the research works that have investigated detection of botnet domains. Here AR is Accuracy Rate, while LDR donates Low Deployment Rate of DNSSEC.

| Work | Method | Advantages | Disadvantages |
|------|--------|------------|---------------|
| [12] | ML/domain length | Small feature space | ✓ |
| [22] | DNSSEC | Strict delegation | ✓ |
| [22] | ML/statistical features | Simple | ✓ |
| [22] | Probability/DNS traffic | Simple | ✓ |

Figure 5: An overview of a botnet infrastructure. Attacker (botmaster) remotely abuses infected devices (zombies) and its connections through C2 channels for malicious activities.

Jin et al. [26] have investigated the characteristics of the DNS log of botnet domain resolution and extracted six different features, e.g., number of source IPs, total querying per day, and querying per hour, among others. These features are then fed into three different classification algorithms, including Adaboost, DT, and NB to detect botnet domain names. In addition, Chang et al. [108] have performed longitudinal study to identify novel botmaster strategies through analyzing multiple active botnet families.

Truong and Cheng [12] have studied DNS traffic to design a detector that distinguishes domain names generated by legitimate users and pseudo-random domain names generated by botnets, such as Conficker [109] and Zeus [110]. The proposed detector classifies the domain names based on machine learning algorithms, e.g., RF, KNN, SVM, and NB, and using extracted features from DNS traffic, including length of domain names and their expected values.

In addition, Xu et al. [29] have carried out a comprehensive study to investigate the feasibility of DNS as a stealthy botnet C&C channel. Therefore, they have tested various strategies, e.g., piggybacking query strategy and exponentially distributed query strategy, that can be utilized at the network level to effectively hide malicious DNS activities. Finally, a countermeasure is proposed that compares the probability distributions of legitimate DNS traffic and tunneling traffic.

Discussion. Although various research works have studied botnets and possible solutions to defend against botnet attacks, such attacks are still compromising users and systems security, highlighting the limitations of the proposed methods. These methods and their strengths and weaknesses are summarized in Table 4. The shortcomings of the proposed methods can be summarized as follows: Low detection accuracy [12, 27], high false alarm rate [29], and low deployment rate of DNSSEC [27]. In addition, the emergence and rise of IoT and associated vulnerabilities opens a new research direction that requires further investigations. For example, the Mirai botnet attacks insecure IoT devices, e.g., routers and digital video recorders [30, 31] highlighting the need for security solutions optimized for the IoT devices [32, 33].
3.4. Phishing Attacks and Defenses

Attackers in phishing attack, attempt to acquire personal and secret information, such as passwords and details of credit cards, by disguising as a trustworthy entity in an electronic communication. In light of the MCSI review published on 2014, the yearly overall effect of the phishing and different types of data fraud attacks has been assessed at as high as US$5 billion [111] only at one particular country, Singapore. Therefore, researchers have conducted a large number of works to detect and mitigate phishing attacks [39, 40, 41, 42, 43].

For example, Bilge et al. [39] have presented a malicious domain detection system, called EXPOSURE, which analyzes various characteristics of DNS names and the way that they are queried. EXPOSURE analyzes DNS traffic and extracts different types of features, including time-based feature, DNS answer-based features, TTL value-based features, and domain name-based feature. Those features are then used by EXPOSURE to design a decision based classifier for automatic detection of a wide range of malicious domain names, e.g., botnet C&C servers, phishing sites, and scam hosts.

Web-based approaches. Cook et al. [42] proposed a mechanism, Phishwish, to detect phishing messages or emails with a small false positive rate. The idea of Phishwish was to provide better protection against zero-hour attacks than blacklists. Phishwish analyzes text and HTML formatted emails using four general rules: identification and analysis of the log-in URL in the email, analysis of the email headers, analysis of the images in the email, and analysis of the accessibility of the URLs.

Medvet et al. [43] presented a phishing detection method based on the visual-similarity that works based on analyzing three main characteristics of websites; the text pieces, the embedded images in the page, and the overall visual appearance of the website. A similarity signature is calculated for legitimate and suspicious websites and an alarm is raised if the signatures were too similar. Furthermore, Hara et al. [44] have proposed a phishing detection technique based on visual similarity that works even if the original website is not registered in the database. A collection of legitimate websites are used to train the classifier and store in a database. A suspected website’s snapshot is then compared to the websites in the database; the suspected website is labeled as phishing if the image similarity metric is above a certain threshold. In case where there is no similar website in the database the suspected website is considered as legitimate website.

Discussion. Despite the large number of research works that have studied the detection and mitigation of phishing attacks, the landscape of phishing is an expanding area contributing very much to the DNS security landscape. Table 5 summarizes the proposed methods in the literature and their strengths and weaknesses. Low detection accuracy [42] [44], requiring large number of train samples [39], dependency of the performance to train inputs [39] [43], and not being adaptive to the changes in the scenarios [44] are some of the shortcomings of the proposed methods in the literature. In addition, phishers are targeting mobile messaging and social apps in increasing rate [45], which requires further detailed investigations to be addressed.

3.5. DNS Manipulation

The process of diverting legitimate DNS requests to malicious IP addresses, which are hosted on misbehaving servers, is known as DNS manipulation. DNS manipulation behavior of attackers exposes users to threats, such as phishing and content injection. It is thus not a surprise that a large number of studies have focused on DNS manipulation [35, 36, 37, 38].

For instance, the vulnerability of the user-side DNS infrastructure to record injection threats have been measured by Schomp et al. [35]. Their analysis showed that a large number (9%) of open DNS resolvers are vulnerable to record injection attacks and are abused by attacks on shared DNS infrastructure. They have measured the extent of popular record injection attacks, e.g., Kaminsky [112] and the deployment of familiar defensive methods, e.g., 0x20 encoding [94].

Kuhrer et al. [36] have investigated the negative aspects of open DNS resolvers that can be attacked by attackers for various malicious activities, e.g., amplification DDoS attacks, DNS manipulation, and cache poisoning, among others. The authors have investigated the response authenticity of the open resolvers from the user’s point of view, and found that millions of them deliberately manipulate DNS resolutions for censoring communication channels, injecting advertisements, serving malicious files, and performing phishing attacks. Furthermore, Jones et al. [113] have proposed an approach for the detection of unauthorized DNS root manipulation using two different techniques. The first technique analyzes the latency to root servers and the second technique analyzes route hijacks. Their analysis revealed that the entities that are operating unauthorized root servers can completely control the entire Internet name space for any system within their sphere.

Pearce et al. [37] proposed a scalable and lightweight system, called Iris, which measures and detects the widespread of DNS manipulation at the scale of countries, which were manipulated based on tactics that rely on DNS resolvers. Iris collects DNS queries through geographically distributed DNS resolvers and analyzes the responses based on two metrics, the consistency and the independent verifiability metrics. Iris investigates both sensitive domains and domains of popular websites for DNS manipulation. Their analysis showed that DNS manipulation is a phenomenon that is heterogeneous across resolvers, domain names, and countries.

Trevisan et al. [38] proposed an automatic and parameter-free detection system, called REMeDy, which identifies rogue DNS resolvers. REMeDy analyzes DNS traffic and evaluates
the consistency of responses across all resolvers for automatic identification of manipulated responses.

**Discussion.** DNS infrastructure suffers from multiple issues, e.g., inefficient look-ups, unreliability of IP-level fragmentation, and ISP-driven DNS manipulation [34]. Moreover, record injection vulnerabilities, e.g., Kaminsky [112] are widespread [35]. Attackers can utilize standard-conforming resolvers to circumvent mitigation techniques e.g., source port randomization as well as IP address randomization [23, 38] to manipulate DNS. In addition, ISPs are clearly willing to involve in DNS manipulation for reasons, such as error traffic monetization [75]. Furthermore, it has been demonstrated that root servers can be abused by unauthorized entities for malicious activities, such as blocking access to websites or manipulate responses through MITM proxies [113].

### 3.6. Malicious Domains Detection

Domains and domain names provide a hierarchy of unique identifiers that guide traffic across the Web and identify websites, servers and other resources. Notwithstanding, in the form of malicious domains, they are an essential apparatus in the hands of miscreants. There are no silver bullets for securing against malicious domains. Nonetheless, understanding domain names can enable firms and individual stakeholders to protect themselves against attacks. A wide variety of research works have been done for accurate detection of malicious domain names [37, 59, 48, 49, 50, 51, 52, 53, 26, 55, 56, 57, 58].

This work is classified into two general categories, classification based [36, 46, 47, 59, 48, 26] and inference based [51, 55, 58]. The classification based approaches rely on local network and host information; however, inference based approach exploits the global information along with local information. The general structure of the malicious domain detection systems based on classification approaches is shown in Figure 6 **Classification based approach.** Antonakakis *et al.* [46] have presented a malicious domain detection system, Notos, which works based on assigning dynamic reputation score to any domain name in a DNS query. Notos maintains up-to-date DNS information of domain names by gathering such information from various sources such as the DNS zone and associated IP addresses, BGP prefixes, AS information, and honeypot analysis. This information is then used to build models of benign/malicious domain names, and then used to calculate the reputation scores of new domains indicative of their maliciousness.

Instead of monitoring traffic from local recursives, Antonakakis *et al.* [47] presented Kopsis, which detects malware-related domains by monitoring DNS traffic at the upper DNS hierarchy. Kopsis offers a new set of traffic features, including daily patterns of domain name resolution, the significance of requesters within each epoch, and the reputation of IP addresses that leverage a global visibility, thus leading to an early detection of malicious domains. Kopsis offers independent malware domain detection within DNS operator’s authority and works well even if there is no available information about IP reputation.

Bilge *et al.* [39] have presented a malicious domain detection system, called EXPOSURE, which analyzes a large set of passive DNS records. EXPOSURE analyzes various characteristics of DNS names and the way that they are queried, and extracts various kinds of features, namely time-based feature, DNS answer-based features, TTL value-based features, and domain name-based feature. These features are then used to design a decision tree based classifier that automatically detects a wide range of malicious domain names, e.g., botnet C&C servers, phishing sites, and scam hosts. Later they extended their work by analyzing a large set of real-world DNS requests over 17 months period of EXPOSURE’s operation [48]. They showed that EXPOSURE is able to make meaningful connections between various malicious domains. For example, they exist IP addresses that are shared among thousands of unique domain names indicative of botnet-related domain names.

Jin *et al.* [26] have investigated the characteristics of the DNS log of botnet domain resolution to extract representative features, e.g., Number of source IPs, Total querying per day, Querying per hour etc. These features are then used to build a malicious domain detection system that consists of three different classification algorithms: Adaboost, DT, and NB.

**Inference-based approach.** Strong associations of domains with known malicious domains is a vital point in effective malicious domain detection. Therefore, Khalil *et al.* [54] have designed an association-based scheme for the detection of malici-
cious domains with high accuracy and coverage. They have analyzed active DNS data and extracted two types of features, namely, domain based features and IP block based features. These features are then used to accurately distinguish public IPs from dedicated ones, consequently building high-quality associations between domain names and identifying malicious domains. Furthermore, Gao et al. [51] have utilized temporal correlation analysis of DNS queries to detect unknown malicious domains based on the related known malicious anchor domains. The proposed approach is able to detect a wide range of correlated malicious domain groups, e.g., phishing, spam, and DGA-generated domains. The results demonstrated that on average each of the known malicious anchor domains can detect more than 53 previously unknown malicious domains.

Luo et al. [55] have proposed a comprehensive framework that uses evolutionary learning for detection of diverse clusters of DNS failures, e.g., highly random, mutated string, and substring domain name failure patterns using different syntactic and temporal patterns. In order to reduce the computational cost, less suspicious clusters were removed while more suspicious cases were preserved for further analysis.

**Time-of-registration detection.** Early detection of potentially malicious DNS domains are of high importance for network operators, registries or registrars, and law enforcement professionals to defend against many Internet attacks. Therefore, Hao et al. [53] have explored registration behaviors of malicious domains with an eye towards features that might indicate the maliciousness of a domain at the time of registration. The characteristics of registrars, domains life cycle, and names patterns were investigated and the results showed that spammers use a small set of registrars to register previously used domains in bulk.

Hao et al. [52] have proposed a malicious domain detection system, PREDATOR, that proactively detects malicious domains using time-of-registration statistical features, including registration history features, domain profile features, and batch correlation features. These features are used to build ML-based classifier, e.g., SVM, CMP that detects malicious domains before they appear in DNS blacklists. Moreover, Felegyhazi et al. [50] proposed an approach for proactive malicious domain blacklisting that relies on registration and name server information of small set of known malicious domains to predict whether a larger set of domains are malicious. Their analysis showed that on average 3.5 to 15 new domains can be derived from a given known malicious domain.

**Web-based approach.** Ma et al. [58] have studied application of batch and online learning approaches for detection of malicious web sites using live feed of labeled URLs and based on the lexical and host-based features. Their analysis showed that size of the training dataset and changes on the distribution of the features affects the performance of the batch algorithms. However, online confidence weighted algorithm is able to classify the labeled URLs in real time and with high accuracy. Furthermore, Canali et al. [49] have proposed a malicious web pages detection system based on a fast and reliable filter, Prophiler, that filters out benign web pages from further costly analysis. Prophiler examines a web page for malicious contents using different types of static features, such as HTML features, JavaScript features, URL and host-based features.

**Discussion.** Although there exists a large body of research devoted to detection of malicious domain names, miscreants have used the DNS to build malicious network infrastructure, e.g., botnet [11]. The summary of the proposed methods and their strengths and weaknesses are listed in Table 6. Shortcomings of the proposed methods can be summarized as following: low detection accuracy [46, 49, 26, 50, 52], local visibility [46, 39, 48], expensive computation [56], dependency on training dataset [46, 39, 48, 47], requiring correlation between anchor malicious domain and unknown malicious domains [55, 51], and can be evaded by avoiding certain features or behaviors [59, 48, 52].

3.7. Domain Generation Algorithm

Malware operation sometimes relies on a fixed domain or IP addresses, which can be hindered quickly. Thus, the primary idea of Domain generation algorithms (DGAs) were instead of developing new versions of the malware or setting everything up at another server, the malware would transfer its C&C to another domain at regular intervals. DGAs are utilized to intermittently produce a large number of domain names and usually for malicious purposes. For instance, Figure 8 shows a sample DGA code that is utilized to generate algorithmic domain names. DGAs are being used as the main method of communication in botnets. For example, Figure 7 shows an overview of the communication of malware-infected hosts with C&C servers. There exist several research works that explored the ecosystem of DGAs [59, 60, 61, 28, 12, 62].

Yadav et al. [59] studied algorithmically generated domain names for domain fluxing. Statistical measures, such as Kullback-Leibler divergence [11], Jaccard index, and the Levenshtein edit distance were used to explore whether a group of domains are algorithmically generated or not. They have applied their method to Tier-1 ISP’s DNS traces and the results showed that it can automatically detect domain fluxing, even for unknown and unclassified botnets, e.g., Mjuyh.

Plohmann et al. [28] have conducted a large-scale measurement study to understand domain generating malware. They investigated and presented a taxonomy for 43 DGA-based malware families to identify and compare their characteristics. Using reverse engineering and re-implementation of the algorithms they pre-computed more than 159M possible DGA-based domain names, which can be used for both predictive blocking

![Diagram](image-url)
def generate_domain(y, m, d):
    domain =
    for i in range(16):
        y = (y * 8 + y) >> 11
        m = (m * 4 + m) >> 25
        d = (d * (d < 13)) >> 19
        m = m & 0xffffffff
        d = m - chr((y - m * d) >> 25) + 97
    return domain

3.8. Domain Name Squatting

Domain name squatting, also known as cybersquatting, is registering or using an Internet domain name with the bad intent to profit from a trademark belonging to someone else. Cyber-squatters register variants of popular trademark names through different squatting strategies, including typosquatting, bit-squatting, combosquatting, and soundsquatting. Domain name squatting practices expose users to a variety of vulnerabilities, such as trademark infringement, monetization, malware, and scams. There exists several research works that attempted to investigate the landscape and impact of the domain squatting [65, 67, 41, 68, 66, 65, 72], which we define below.

3.8.1. Typosquatting

Typosquatting is one of the most common forms of domain squatting that targets Internet users who incorrectly type a website address into their web browser, e.g., www.example.com instead of www.example.com, and mainly for monetization activities. To this end, researchers have attempted to systematically introduce typo-generation models [63, 64]. For instance, Wang et al. [63] have discovered common typo-generation models of Alexa top domain list, namely missing-dot typos, character-omission typos, character-permutation typos, character-substitution typos, and character-duplication typos. Similarly, Banerjee et al. [64] have suggested different methods for generating typosquatting domains, including 6.1-mod-inplace, 7.1-mod deflate, and 8.1-mod-inflate. Furthermore, Spaulding et al. [8] have explored the landscape of the typosquatted domain names and summarized common typo-generation models and monetization strategies, which are shown in Figure 9.

Achten et al. [65] conducted a longitudinal study of typosquatting domains and trends. Their analysis showed that 95% of the studied top domains are targeted by typosquatters; however, only few trademark owners take protection practices, such as proactive registration of typosquatting domains. Furthermore, they observed a change in the trend and monetization behaviors of typosquaters over time and through hosting different types of websites, e.g., parked/ads/for sale domains and scams. Moreover, Szurdi et al. [66] conducted a large-scale measurement of typosquatting domain registrations in the .com TLD. This study explores the monetization strategies among less known domains. Their analysis showed that 95% of typosquatting domains target the long tail of the popularity distribution. In addition, they found that typosquatting domains constitute 20% of the domains in the .com TLD and the trend is increasing. Moreover, et al. [64] revealed that for a given authoritative domain the percentage of active typosquatting domains decreases remarkably as the popularity of domains decline.

However, the results of these two research works [66, 65] are in contrast to the results presented by Khan et al. [67] who have explored the impact of typosquatted domains on the users through measuring the harm experienced by users using intent inference metric. Their analysis showed that typosquatting practice increases the time it takes to find an intended website, generally 1.3 seconds per typosquatting event over the alternative of receiving a browser error page.

Spaulding et al. [68, 69] have conducted a user study to understand the effectiveness of the typosquatting strategies in deceiving users. The behaviour of users, who have been exposed to several typosquatting URLs, have been examined in an attempt to figure out whether their behavior is improved by security education and increasing their awareness about typosquatters. They found that user’s behavior is affected the most by factors such as age and education. Furthermore, the results showed that typo-generation models, such as permutations and substitutions of characters are among the most effective techniques in deceiving users, as opposed to techniques that rely on adding characters to the domain names.
3.8.2. Bitsquatting

The main premise of bitsquatting [71] is hardware malfunction and it refers to registration of domain names with one different bit from popular domain names in an attempt to receive unintentional traffic caused by bit-flip errors in the memory of computers. This theory is tested, for the first time, by Dinaburg [70] through investigating all HTTP requests to 30 registered bitsquat versions of popular domain names, e.g., amazon.com. The results of eight months of experience showed that there were over 52K bitsquat requests from more than 12K unique IP addresses, highlighting the feasibility and presence of bitsquatting. Finally, a hardware-based defense strategy that analyzes the integrity of data stored in hardware is introduced. Furthermore, Nikiforakis et al. [71] have explored the bitsquatting phenomenon in more detail and found that attackers register new bitsquatting domains daily. Their results showed that bitsquatting domains involve different monetization practices, such as parked/ads/for sale domains, affiliate marketing, and scams. Finally, they reviewed possible defense strategies against bitsquatting, such as data integrity validation, DNSSEC, and preregistering all possible bitsquatting domains.

Vissers et al. [72] have conducted a large-scale analysis of configuration issues, e.g., typosquatting and hardware errors, e.g., bitsquatting over 10K popular name server domains. Their analysis confirmed the presence of squatting vulnerabilities among name servers, thus compromising all domains that rely on them. Furthermore, they analyzed the security practices of popular name servers and found that 12K domains are directly exposed to being hijacked through configuration errors related to their name server and 52.8M domains are being targeted by name server bitsquatters. Finally, they suggested deployment of DNSSEC as an effective countermeasure.

3.8.3. Combosquatting

Combosquatting is a type of domain squatting in which a popular trademark is combined with one or multiple phrases, e.g., www.youtube-login.com instead of www.youtube.com. In order to explore its scope, Kintis et al. [41] have analyzed lexical characteristics and temporal properties of DNS records, collected from PDNS and ADNS sources over six years, for in-depth and long-term analysis of combosquatting domains. The results showed that combosquatting domains are generated by adding a single token to the original domain and they are far more prevalent than typosquatting. In addition, they found that combosquatting domains are increasing in numbers, while a large fraction of them (≈60%) stay active for more than 1000 days.

3.8.4. Soundsquatting

Soundsquatting is a type of domain squatting technique that takes the advantage of homophones words and the users confusion, such as such as whether, weather and idle, idol, idyll to generate squatted domains. Nikiforakis et al. [73] have explored the landscape of soundsquatting and showed that attackers are already familiar with soundsquatting concepts, e.g., generative models of soundsquatted domains and are monetizing them in different ways, such as parked/ads/for sale domains, hit stealing, and affiliate marketing. Furthermore, they proposed a tool, Auto Sound Squatter (AutoSS), that automatically generates soundsquatting domains. In addition, they showed that users are exposed to soundsquatting through the abuse of text-to-speech software, including the built-in screen reader of Windows XP, Windows 7 and Mac OS X, the Thunder screen reader [115], ORCA [116], and Skyvi [117].

Discussion. Researchers have identified various domain squatting strategies, e.g., typosquatting, bitsquatting, etc., which are listed in Table 7 along with the summary, contribution, and limitations of the research works that investigated them. Despite the proposed methods to mitigate domain squatting, squatting phenomenon is continuing to thrive and expand [66, 41]. The limitations of the existing research proposals can be summarized as following: 1) lack of practical yet automatic domain squatting detection system [41], 2) requiring a huge amount of user traffic or content for building models [67], and 3) lack of practical yet effective mitigation techniques [71, 72, 70]. For instance, the research proposals which suggest deployment of DNSSEC [71, 72] should consider the fact that deployment rate of DNSSEC although increasing but still far from ideal point.

3.9. Parked Domain Monetization

Domain parking refers to the registration of an Internet domain name without having the domain associated with any services such as e-mail or a website. Domain parking can be classified as monetized and non-monetized. In the former, advertisements are shown to visitors and the domain is monetized;
Table 7: Summary of the works that have studied the landscape of the domain name squatting. T= Typosquatting, B= Bitsquatting, C= Combosquatting, SS= Soundsquatting, PAF= Parked/Ads/For sale domains, S= Scams, HS= Hit Stealing, and AM= Affiliate Marketing.

| Work | Method | Monetization strategy | Contribution | Disadvantages |
|------|--------|-----------------------|--------------|---------------|
| [5]  | ✓      | ✓                     | Squatting comparison | —             |
| [41] | ✓      | ✓                     | Combusquatting   | No detection  |
| [63] | ✓      | ✓                     | Typo-generation models | Limited data |
| [65] | ✓      | ✓                     | Longitudinal study | Limited data  |
| [66] | ✓      | ✓                     | YATT mitigation tool | Updating blacklist |
| [67] | ✓      | ✓                     | Typosquatting detector | Large user traffic |
| [70] | ✓      | ✓                     | Bitsquatting/defense | impractical   |
| [71] | ✓      | ✓                     | Bitsquatting/defense | Costly/impractical |
| [72] | ✓      | ✓                     | Name server squatting | Low deployment |
| [73] | ✓      | ✓                     | Soundsquatting    | Accuracy rate  |

whereas, non-monetized parked domains reserve the domain name for future development or to protect against the possibility of cybersquatting. Despite technically being a legitimate business, parked domain monetization can get mixed up with suspicious practices and malicious contents such as malware. Therefore, several works were conducted to understand parked domains and monetization strategies [74, 66, 75, 76].

Alrwais et al. [74] carried out a comprehensive study on the dark side of the parked domains to understand monetization, its scope, and magnitude. They controlled the start and end node of the monetization process, using an infiltration method, to send crawling traffic among specified start and end nodes with the monetization entities in between. Their analysis of one thousand seed redirection chains proved the presence of threats, such as click fraud, traffic spam, and traffic stealing in the monetization process. In addition, the results of this study showed that the revenue of the responsible parties for illicit monetization activities is as high as 40% of the total revenues.

Szurdi et al. [65] have conducted a large-scale measurement study to understand the impact of typosquatting on the monetization strategies of attackers among less known domains. Their analysis showed that large proportion (95.0%) of the typosquatting domains are targeting less popular websites for domain monetization techniques e.g., parked ads. Furthermore, Weaver et al. [75] have investigated the error traffic monetization practices of ISPs through analyzing 66,000 Netalyzr session traces. Their analysis showed that ISPs were clearly involving on error monetization techniques of ads redirection chains proved the presence of threats, such as click fraud, traffic spam, and traffic stealing in the monetization process. In addition, the results of this study showed that the revenue of the responsible parties for illicit monetization activities is as high as 40% of the total revenues.

Vissers et al. [76] have investigated the ecosystem of parked domains from security view point to understand the security risks of parked domains on users. Their analysis showed that parked domains expose users into a set of threats, e.g., malware, scam, and inappropriate content. Finally, they have presented a RF-based parked domain detection system through analyzing a set of generic features, including HTML features, HTTP archive features, frame Features, and domain name features.

Discussion. Although domain parking is technically a legitimate business, it has been mixed up with malicious practices such as click fraud, malware, scam, malicious contents, etc. [76] Dishonest parking services offer high revenue for responsible parties, which may compromise the benefits of legitimate advertisers and traffic buyers, by manipulating the traffic and redirecting them to rogue ad servers [74]. Moreover, domain monetization is prevalent among both popular domains and less known domains [66]. In addition, ISPs actively involve traffic monetization activities [75].

3.10. Privacy Leakage

A range of works have been conducted to address privacy-related issues of DNS [122, 77, 121, 78, 123, 79, 120, 124, 125, 126]. For example, Herrmann et al. [77] have explored the emerging threat of third-party DNS resolvers, e.g., Google Public DNS and OpenDNS to online privacy and introduced a lightweight privacy-preserving name resolution service called EncDNS. EncDNS is designed based on the encryption of DNS requests in standards-compliant DNS messages. EncDNS is compatible with existing popular DNS resolvers and offers low-latency DNS resolution. Furthermore, Shulman [127] have explored the existing proposals on DNS privacy that suggested encryption of DNS requests and introduced a lightweight privacy-preserving name resolution service called EncDNS. EncDNS is designed based on the encryption of DNS requests in standards-compliant DNS messages. EncDNS is compatible with existing popular DNS resolvers and offers low-latency DNS resolution. Furthermore, Shulman [127] have explored the existing proposals on DNS privacy that suggested encryption of DNS requests as a solution. The results demonstrated that a straightforward application of encryption alone may not provide the desired DNS privacy protection.

Krishnan et al. [121] have explored the privacy implications of DNS prefetching. Their analysis showed that an adversary can abuse the context inserted to a resolver’s cache through prefetching to launch disclosure attacks, e.g., reconstructing searched terms. Thus, they suggested that DNS prefetching should be turned off by default.

Zhu et al. [120] have presented a connection-oriented DNS, called T-DNS, to address the privacy and security issues of the connectionless DNS. T-DNS takes advantage of both TCP and TLS: while TCP protects the server against amplification attacks, TLS defends against eavesdroppers to the RDNS resolvers. In addition, Reddy et al. [128] have explored the protection of the privacy-sensitive information of DNS queries and responses. They presented a protection mechanism, called the Datagram Transport Layer Security (DTLS), for DNS exchange. DTLS counters passive listening and active attacks. The pro-
posed mechanism reduces the DTLS round trips and the handshake size as well.

Zhao et al. [118] have investigated the privacy disclosure of DNS queries and proposed a privacy-preserving query scheme, called Range Query, which reduces privacy disclosure by concealing the actual queries using noisy traffic. However, Castillo-Perez and Garcia-Alfaro [122] demonstrated that the privacy ensured by noisy traffic is not only difficult to analyze, but also does introduce undesired latency and bandwidth consumption. Castillo-Perez and Garcia-Alfaro [122] have evaluated two DNS privacy-preserving proposals, the Range Query [118] and the Privacy Information Retrieval (PIR) schemes [119]. They demonstrated that both of these approaches are not desired. The first approach increases the latency and the bandwidth during the execution and resolution of queries. Although the second approach is designed to address the limitations of the first approach, e.g., bandwidth consumption, its functionality requires major modification of the DNS protocol, and relies on the use of DNSSEC.

Lu and Tsudik [78] explored DNS privacy leaks during domain resolution and presented Privacy-Preserving DNS (PPDNS) that mitigates the privacy issues in DNS. PPDNS takes advantage of both distributed hash tables (DHTs) and computational private information retrieval (cPIR). While DHT is an alternative naming infrastructure that provides name resolution query privacy, cPIR reduces communication overhead. However, Federrath et al. [129] argue that although PPDNS provides high level of privacy, it would not be adopted in the near future due to its computational complexity and requirement for completely different DNS infrastructure.

DNS blocking of unintended queries is among the suggested solutions to improve the DNS privacy. For example, Appelbaum and Muffett [130] have studied Tor’s privacy and suggested to block .onion names at stub, recursive, and authoritative resolvers to improve it’s privacy. Mohaisen and Ren [125] have investigated the leakage of .onion at the A and J DNS root nodes over a longitudinal period of time and have found that .onion leakage is common and persistent at DNS infrastructure. Furthermore, Mohaisen et al. [126] have studied the impact of blocking of unintended queries under different adversarial settings on the DNS privacy. The results highlighted that partial blocking at stub resolver would negatively affect the DNS privacy under certain adversary models; however, blocking the queries at the recursive would result in favorable privacy outcomes.

**Discussion.** Full and proper deployment of DNSSEC relies on cooperation between domain owners, name server owners, registries, and ISPs. Thus, various methods have been proposed to improve the privacy of DNS which are listed in Table 8 along with their strengths and weaknesses. It should be noted that DNSSEC does not necessarily protect privacy. Although DNS over TLS protects against eavesdropping, it is not clear what of privacy it offers. In addition, existing approaches require modifications to the DNS protocols [118]. Privacy issues have been overlooked by DNS security efforts (such as DNSSEC) and are thus likely to propagate into future versions of DNS [78], requiring further attention.

### 4. DNS Research Methods

This section is devoted to investigating various data analysis methods, utilized to detect, model, and mitigate the aforementioned DNS threats. As it is outlined in Figure 10, the conducted research works will be categorized under data collection methods, data analysis techniques, and the scope of the analysis. This categorization will help us to better understand the advantages and disadvantages of different methodologies in the DNS security and privacy landscape, which will be an asset for future investigations. The topical classification of the DNS research methods, with sample works, is presented in Table 14.

#### 4.1. Data Collection Methods

This section will describe two main DNS data collection methods, the passive DNS (PDNS) data and active DNS (ADNS) data, and the associated works.

**4.1.1. Passive DNS**

Florian Weimer [135] has invented “Passive DNS” or “Passive DNS replication” technology to opportunistically reconstruct a partial view of the data available in the global DNS into a central database for further investigations. End-user’s interactions can be monitored as they happen using passive monitoring [136]. High-level architecture of passive DNS measurement systems is shown in Figure 11. Passive DNS databases based on valuable information they collect, have been considered as an invaluable asset of cybersecurity researchers to combat a wide range of threats such as malware, botnets, and malicious actors [80, 47, 81, 82, 83, 12, 11, 84, 48, 41, 67, 22, 51].

| Work | Method     | strengths                          | Weaknesses                        |
|------|------------|------------------------------------|-----------------------------------|
| 77   | EncDNS     | Lightweight/low-latency            | Unresolvable internal domains     |
| 78   | PPDNS      | Improved privacy                   | Complexity/DNS modification       |
| 79   | DLV-Aware/PP-DLV | Improved privacy | MITM attacks/record modification  |
| 118  | Range query| Improved bandwidth/Simple/flexible | Bandwidth/response latency         |
| 119  | PIR        | Improved bandwidth                 | Protocol modification             |
| 120  | T-DNS      | Improved privacy/security           | Response latency                  |
| 121  | No DNS pref | Inference attacks                  | Inaccurate TTL/Unstable profiles   |
Table 9: Topical classification threats addressed in the literature, with sample work. MD=Malicious Domains, CP=Cache Poisoning, PH=Phishing, MA=Manipulation, AD=Amplification/DoS, DG=Domain Generation, B=Botnet, DS=Domain Squatting, M=Monetization, PL=Privacy Leakage.

| Work | MD | CP | PH | MA | AD | DG | B | DS | M | PL |
|------|----|----|----|----|----|----|---|----|---|----|
| 1    |    |    |    |    |    |    |   |    |   |    |
| 69   |    |    |    |    |    |    |   |    |   |    |
| 7    |    |    |    |    |    |    |   |    |   |    |
| 91   |    |    |    |    |    |    |   |    |   |    |
| 10   |    |    |    |    |    |    |   |    |   |    |
| 11   |    |    |    |    |    |    |   |    |   |    |
| 12   |    |    |    |    |    |    |   |    |   |    |
| 13   |    |    |    |    |    |    |   |    |   |    |
| 16   |    |    |    |    |    |    |   |    |   |    |
| 17   |    |    |    |    |    |    |   |    |   |    |
| 18   |    |    |    |    |    |    |   |    |   |    |
| 19   |    |    |    |    |    |    |   |    |   |    |
| 20   |    |    |    |    |    |    |   |    |   |    |
| 21   |    |    |    |    |    |    |   |    |   |    |
| 22   |    |    |    |    |    |    |   |    |   |    |
| 23   | √  |    |    |    |    |    |   |    |   |    |
| 26   |    |    |    |    |    |    |   |    |   |    |
| 27   |    |    |    |    |    |    |   |    |   |    |
| 28   |    |    |    |    |    |    |   |    |   |    |
| 29   |    |    |    |    |    |    |   |    |   |    |
| 34   |    |    |    |    |    |    |   |    |   |    |
| 35   |    |    |    |    |    |    |   |    |   |    |
| 36   |    |    |    |    |    |    |   |    |   |    |
| 37   |    |    |    |    |    |    |   |    |   |    |
| 38   |    |    |    |    |    |    |   |    |   |    |
| 40   |    |    |    |    |    |    |   |    |   |    |
| 41   |    |    |    |    |    |    |   |    |   |    |
| 42   |    |    |    |    |    |    |   |    |   |    |
| 43   |    |    |    |    |    |    |   |    |   |    |
| 46   |    |    |    |    |    |    |   |    |   |    |
| 47   |    |    |    |    |    |    |   |    |   |    |
| 48   |    |    |    |    |    |    |   |    |   |    |
| 49   |    |    |    |    |    |    |   |    |   |    |
| 50   |    |    |    |    |    |    |   |    |   |    |
| 51   |    |    |    |    |    |    |   |    |   |    |
| 52   |    |    |    |    |    |    |   |    |   |    |
| 53   |    |    |    |    |    |    |   |    |   |    |
| 54   |    |    |    |    |    |    |   |    |   |    |
| 55   |    |    |    |    |    |    |   |    |   |    |
| 56   |    |    |    |    |    |    |   |    |   |    |
| 57   |    |    |    |    |    |    |   |    |   |    |
| 58   |    |    |    |    |    |    |   |    |   |    |
| 66   |    |    |    |    |    |    |   |    |   |    |
| 69   |    |    |    |    |    |    |   |    |   |    |
| 70   |    |    |    |    |    |    |   |    |   |    |
| 79   |    |    |    |    |    |    |   |    |   |    |
| 82   |    |    |    |    |    |    |   |    |   |    |
| 83   |    |    |    |    |    |    |   |    |   |    |
| 84   |    |    |    |    |    |    |   |    |   |    |
| 88   |    |    |    |    |    |    |   |    |   |    |
| 89   |    |    |    |    |    |    |   |    |   |    |
| 99   |    |    |    |    |    |    |   |    |   |    |
| 113  |    |    |    |    |    |    |   |    |   |    |
| 120  |    |    |    |    |    |    |   |    |   |    |
| 121  |    |    |    |    |    |    |   |    |   |    |
| 122  |    |    |    |    |    |    |   |    |   |    |
| 123  |    |    |    |    |    |    |   |    |   |    |
| 124  |    |    |    |    |    |    |   |    |   |    |
| 125  |    |    |    |    |    |    |   |    |   |    |
| 126  |    |    |    |    |    |    |   |    |   |    |
| 131  |    |    |    |    |    |    |   |    |   |    |
| 132  |    |    |    |    |    |    |   |    |   |    |
| 133  |    |    |    |    |    |    |   |    |   |    |
| 134  |    |    |    |    |    |    |   |    |   |    |

Table 10: Topical classification of the DNS threats addressed in the literature, cont. of Table 9.

| Work | MD | CP | PH | MA | AD | DG | B | DS | M | PL |
|------|----|----|----|----|----|----|---|----|---|----|
| 78   |    |    |    |    |    |    |   |    |   |    |
| 79   |    |    |    |    |    |    |   |    |   |    |
| 82   |    |    |    |    |    |    |   |    |   |    |
| 84   |    |    |    |    |    |    |   |    |   |    |
| 88   |    |    |    |    |    |    |   |    |   |    |
| 89   |    |    |    |    |    |    |   |    |   |    |
| 99   |    |    |    |    |    |    |   |    |   |    |
| 113  |    |    |    |    |    |    |   |    |   |    |
| 120  |    |    |    |    |    |    |   |    |   |    |
| 121  |    |    |    |    |    |    |   |    |   |    |
| 122  |    |    |    |    |    |    |   |    |   |    |
| 123  |    |    |    |    |    |    |   |    |   |    |
| 124  |    |    |    |    |    |    |   |    |   |    |
| 125  |    |    |    |    |    |    |   |    |   |    |
| 126  |    |    |    |    |    |    |   |    |   |    |
| 131  |    |    |    |    |    |    |   |    |   |    |
| 132  |    |    |    |    |    |    |   |    |   |    |
| 133  |    |    |    |    |    |    |   |    |   |    |
| 134  |    |    |    |    |    |    |   |    |   |    |

Figure 10: DNS Research Methods. We have categorized DNS research method into data collection methods, data analysis techniques, and analysis scope.

For example, Callahan et al. [81] have analyzed 200 million DNS queries collected through passive monitoring of DNS traffic of 90 home residential neighborhood network in the U.S. [137] in order to investigate the evolution and the behavior of the DNS servers and clients in modern DNS systems. Moreover, Gao et al. [51] have analyzed 26 billion DNS query-response pairs collected by Security Information Exchange (SIE), now part of DNSDB [138], from more than 600 recursive DNS resolvers distributed over North America and Europe, to empirically re-examine the performance and the operational characteristics of the DNS infrastructure. In addition, they found that temporal correlation analysis of the collected passive DNS queries can be used to detect malicious domain names.

Hao and Wang [22] have studied 1.2 billion trace logs of outgoing DNS queries collected at local DNS servers at the College of William and Mary (WM) and the University of Delaware (UD) to understand the impact of one-time-use domain names.
on the performance of DNS caching. In addition, Alrwais et al. \[80\] have analyzed 1.5 TB of passive DNS data collected by monitoring 25 snapshots of the entire IPv4 address space to systematically study the trend of BulletProof Hosting (BPH) services and potential detection and mitigation methods.

Liu et al. \[58\] have queried passive DNS data from Farsight Security passive DNS database, DNSDB \[138\], and the biggest public passive DNS system in China \[139\] to detect malicious domains. Moreover, Tajalizadehkhoob et al. \[82\] have analyzed 1.259 shared hosting providers extracted from DNSDB to explore the distribution of web security features and software patching practices in shared hosting providers.

Furthermore, Khan et al. \[67\] have utilized passive DNS data to study the negative impact of the typosquatting on the users through the time it takes to find the intended website. Kintis et al. \[41\] have analyzed 13.1 billion passive DNS RRs collected from the largest ISP in the U.S., to understand how combosquatting is used by miscreants for malicious purposes.

Bilge et al. \[48\] have analyzed the passive DNS data obtained by SIE to detect and block malicious domain names in real time. In addition, Li et al. \[84\] have investigated the topological relation of hosts and malicious websites using the same PDNS API, SIE. Moreover, Perdisci et al. \[11\] have utilized the SIE PDNS traffic to detect and track malicious domain-flux networks. Similarly, Truong et al. \[12\] have analyzed passive DNS traffic for detection of domain-flux botnets within a monitored LAN network.

Although passive DNS data is widely used to study and address to various vulnerabilities of DNS, e.g., malware, botnets, domain squatting, etc. \[51\], \[22\], \[67\], \[14\], the collection of such data is costly, challenging, and needs restrictive legal agreements. \[83\].

### 4.1.2. Active DNS

The concept of active DNS measurement has been presented to tackle the potential barriers of DNS security research. Active DNS measurement cannot be a replacement for passive DNS measurement; however, it offers several advantages in comparison to its counterpart \[83\]. For example, Internet activities of real users would not fluctuate active measurements. In addition, active DNS measurements could be largely available with least privacy concerns, while it offers an order of magnitude more domain names and IP addresses. Therefore, researchers have developed and used active DNS measurement systems to address DNS security threats \[67\], \[113\], \[83\], \[85\], \[54\], \[41\], \[86\], \[141\]. Generally, active DNS collection systems are composed of four main components: domain seeds, query generator, collection point, and storage. The list of the domains to be queried are gathered in domain seeds. These lists are requested using query generator and the responses are collected at the collection point and then stored in Hadoop cluster for further analysis. High-level architecture of active DNS collection system is shown in Figure 12.

Kountouras et al. \[83\] have presented an active DNS measurement called Thales, which reliably queries, collects, and distills active DNS datasets. A set of large-scale seed domain list including public blacklists \[142\], \[143\], \[144\], the Alexa list \[145\], the Common Crawl dataset \[146\], the domain feed from an undisclosed security vendor, and the zone files for the TLDs consisting of .com, .net, .biz and .org constitute the list of seed domains. Moreover, Van Rijswijk-Deij et al. \[85\] have presented OpenINTEL, an active DNS measurement system that conducts daily measurement for all of the domains in .com, .net, and .org TLDs. OpenINTEL reliably queries and collects over 50% of the global DNS data based on daily active measurements and since February 2015.

Moreover, Khan et al. \[67\] have actively crawled 13.5k suspected domains to study the impact of typosquatting events on both typical users and legitimate websites. Kintis et al. \[41\] have analyzed 455 billion DNS RRs collected daily through the active DNS project \[147\], \[83\] from November 2015 to August 2016, to investigate how attackers utilize combosquatting to conduct variety of malicious activities including trademark infringement, malware, monetization, and phishing. Moreover, Khalil et al. \[54\] have conducted an in-depth analysis of active DNS dataset provided by Thales \[147\], \[83\] to design an association-based scheme for detection of malicious domains with high accuracy and coverage.

Chung et al. \[86\] have collected all DNSSEC-enabled domain names, under the .com, .org, and .net TLDs through OpenINTEL active DNS measurement system \[83\], to study the adoption and management of DNSSEC in both authoritative name servers and DNS resolvers. Moreover, Toorn et al. \[148\] have...
studied the detection of Snowshoe spams through active DNS measurement and using machine learning techniques. **Discussion.** Passive monitoring is better suited as a technology to determine what goes wrong with a website or web application after problems have occurred. Passive DNS, though, is rare, difficult to obtain, and often comes with restrictive legal clauses (i.e., non-disclosure agreements). At the same time, laws and regulations against personal identifiable information, the significant financial cost of the passive collection, and storage infrastructure are some of the reasons that make passive DNS difficult. In addition, passive DNS datasets are limited in scope and time. On the other hand, although active DNS datasets offer several unique characteristics, the amount of offered information would be limited. It should be noticed that active DNS measurement highly relies on the domain seeds. The strong associations of domains with known malicious domains in its /24 IP block etc.

**4.2. Data Analysis Methods**

Cybersecurity researchers have employed variety of data analysis techniques to effectively address to DNS threats. For example, the application of machine learning algorithms are widely used [80, 17, 58]. This section introduces some of the prevalent data analysis techniques that have been used in the cybersecurity community, such as machine learning algorithms [80, 17, 46, 53, 22, 54, 55, 56, 87, 88, 89, 76] and association analysis [59, 90, 54, 91, 51].

**4.2.1. Machine Learning Algorithm**

Antonakakis et al. [17] have designed an Machine Learning (ML)-based cache poisoning detection system, called Anax, which detects malicious changes in cached DNS records in real time with 91.9% of detection accuracy rate. Similarly, Hao and Wang [22] have trained Decision Tree (DT) and Random Forest (RF) machine learning algorithms using syntactical features of one-time-use domain names, e.g., domain name string, the length of query name etc. to detect and expel one-time-use domain names from DNS cache in an attempt to increase the performance of DNS caching.

Alrwais et al. [80] have utilized two machine learning algorithms, namely SVM and RF, to build a detection system to identify malicious network blocks using features extracted from the trends of the BPH services. Khalil et al. [54] have designed a malicious domain detection system using RF model and based on the representative features of dedicated IPs, e.g., the number of fully qualified domain names, the number of second level domains in its /24 IP block etc.

Liu et al. [58] have utilized a set of machine learning models including NN, RF, LR, SVM, and NB to automatically identify shadowed domains with 98.5% accuracy rate. In addition, Vissers et al. [76] have proposed a parked domain detection system based on RF algorithm that is trained over a set of the generic and robust features of parked domains. Moreover, Li et al. [149] have used TSVM algorithm instead of classical SVM to improve the detection accuracy for phishing webpages.

**4.2.2. Association Analysis**

The strong associations of domains with known malicious domains can be utilized to detect malicious domain names. For
example, Khalil et al. [54] have designed an association-based scheme for detection of malicious domains with high accuracy and coverage. Furthermore, Gao et al. [51] have utilized the temporal correlation analysis of DNS queries to identify a wide range of correlated malicious domain groups, e.g., phishing, spam, and DGA-generated domains based on the related known malicious anchor domains. Yadav et al. [59] have utilized statistical measures such as Kullback-Leibler divergence, Jaccard index, and Levenshtein for domain-flux botnet detection. Gomez et al. [91] have studied the application of visualization for understanding the DNS-based network threat analysis.

Discussion. Despite numerous advantages of machine learning approaches, there are still risks and limitations of using them in operation. The foremost challenge is the acquisition and labeling of relevant data from representative vantage points to maximize insights. Even if the data is collected correctly, capturing DNS traffic results in a very large amount of data to analyze, which would be expensive in term of computation and storage. In addition, the performance of machine learning algorithms is contingent upon their structure and learning algorithms. It should be noticed that selecting improper structure or learning algorithms might result in poor results; thus, it is mandatory to try different algorithms for each problem. Furthermore, the training phase of the algorithm would be a time-consuming process, even if the dataset is small, requiring training heuristics.

4.3. Data Analysis Scope

The original design of the DNS did not consider any security details; instead, it was designed to be a scalable distributed system. As the Internet has grown, malicious actors have found weaknesses, mainly due to lack of DNS records verification, in the DNS system that allows them to launch variety of attacks, e.g., phishing, malware, etc. Thus DNSSEC was created to secure the DNS infrastructure. In this section, we will categorize the research works from security point of view into vanilla DNS [35, 40, 29, 92, 93, 94, 18] and DNSSEC [95, 96, 97, 86, 98, 99, 23, 5, 100]. This will help us to understand the advantages, e.g., improved security and disadvantages, e.g., overheads of the deployment of the DNSSEC.

4.3.1. Vanilla DNS

Although DNS is the largest distributed system, it is vulnerable to multiple threats, such as DNS cache poisoning attacks, amplification attacks, etc.. For example, Schomp et al. [35] have demonstrated that user-side DNS infrastructure is vulnerable to record injection threats. Moreover, Xu et al. [29] have demonstrated that attackers can severely abuse DNS-based stealthy C&C channel to efficiently hide malicious DNS activities. In addition, Schomp et al. [93] have showed that shared DNS resolvers are vulnerable to various forms of attack [99, 35], e.g., fraudulent record injection.

In order to address the security problems of the DNS infrastructure, which are mainly due to the lack of integrity and authenticity of DNS records, the utilization of DNSSEC have been suggested [23, 100].
4.3.2. DNS Security Extension

DNSSEC allows clients to verify the integrity and authenticity of DNS records based on a chain-of-trust. Each zone in DNSSEC is composed of two public and private key pairs, including Key Signing Key (KSK) and Zone Signing Key (ZSK). Public Key Infrastructure (PKI) of the DNSSEC is rooted at the KSK of the DNS root zone. Thus, the validation process of DNS response begins at the root and continues until the record is authenticated. An overview of the DNSSEC chain of trust is shown in Figure 13.

Chung et al. [86] have conducted a measurement study on deployment and management of DNSSEC in both authoritative name servers and DNS resolvers. They found that only a tiny fraction ($\approx 1\%$) of all domains in three top TLDs publish DNSKEY, including .com, .net, and .org. Additionally, their results showed that over 30% signed domains fail to upload DS records, highlighting the failure of authoritative name servers in uploading records for a majority of their domain names. Towards the client side of DNSSEC, Lian et al. [152] have studied the capability of the resolvers and end-users to achieve DNSSEC authentication, and concluded that DNSSEC increases the failure rate of end-to-end resolutions.

To categorize the potential root causes of the DNSSEC misconfigurations, Adrichem et al. [95], have conducted a measurement study to access the impact of misconfigurations on the reachability of zone’s network. Their analysis showed that only 7.93% of gathered domain names from .bg, .br, .co, and .se zones attempted to implement DNSSEC, and over 4% were misconfigured. Furthermore, they observed that the DNSSEC-aware resolvers were unable to reach a large fraction ($\approx 73.86\%$) of misconfigured domains. The impact of DNSSEC misconfigurations on DNS query requests has also been addressed by Deccio et al. [153], and in general, DNSSEC misconfigurations can be broadly classified into four categories: DNSKEY, RRSIG, general DNS failure, and miscellaneous. Figure 14 shows the categories and the subcategories of DNSSEC misconfigurations [95, 153].

DNS operators have a critical role in the deployment of DNSSEC, as they are responsible for the maintenance of DNSKEY and RRSIG records. Chung et al. [98] have studied the impact of DNS operators, e.g., registrar, owner, or third-party DNS operator, on DNSSEC deployment. They have observed that many popular registrars fail to support DNSSEC, and as such, there are only 3 mutual registrars among the top 25 popular, and the top 25 fully deployed registrars.

Discussion. Correct deployment of DNSSEC relies on end-to-end establishment of RRSIGs, a cryptographically valid DNSKEY, and DS records. However, a small fraction of all domains ($\approx 1\%$) in the top three TLDs (.com, .net, and .org) publish DNSKEY [86]. Moreover, incomplete or incorrect deployment of DNSSEC may lead to attacks such as domain hijacking. The threat does not completely mitigate even after full deployment. An adaptive attacker may still succeed in the attack by using a fake name server, that enables off-path traffic analysis and a covert channel [99]. The hazards of DNSSEC misconfiguration are not limited to the failures; misconfigured domains are always at risk of being unreachable from a DNSSEC-aware resolver.

5. DNS Entities Scopes

DNS infrastructure comprises of different entities, including DNS name servers, DNS resolvers, hosting providers, and clients. Each entity is designed to serve a specific task. DNS infrastructure’s performance is contingent upon the collaboration and synchronization of these components. In this section, we survey the prior work to provide deeper insights about the impact of different DNS entities on the overall system. We provide an outline of this section in Figure 15 and in Table 16, we present the topical classification of the DNSSEC entities as addressed in the literature, with sample works.

5.1. DNS Name Servers

A DNS server provides name resolution of names into IP addresses, and vice versa. Extensive work has been published to shed light on the name servers’ landscape [154, 47, 155, 156, 157, 42, 16, 158, 51, 159].

5.1.1. Authoritative Name Server

An authoritative name server is responsible for answering queries related to a particular domain name in a zone. The server response contains actual DNS records of the queried domain name, such as A, CNAME, PTR, etc., which highlight the importance of the server’s performance and security. List of the common query types are reported in Table 12. Hao et al. [154] have investigated the characteristics of DNS, e.g., performance, availability, life-cycle and deployment patterns. In their work, they analyzed three major types of DNS servers namely private, upstream, and hybrid authoritative DNS servers, and showed that a majority of websites host the authoritative DNS servers in upstream services for better performance.
Liang et al. [158] studied the impact of top level DNS servers’ uneven distribution on the end-user latency. They observed that despite good performance of top level DNS servers, the quality of the service was unfair in different continents including Africa and South America. In those regions, the performance was 3–6 times worse than Europe and North America. Furthermore, they made two useful observations: 1. Almost all the root servers in Europe and North America work efficiently. 2. Only F, J, and L roots show low query latency in other continents.

Malicious domains can be identified by the number of name servers that a domain name interacts with over time. For instance, Felegyhazi et al. [50] have utilized the inherent characteristics of domain registrations and their appearance in DNS zone files to build a proactive malicious domain detection system. Mohaisen et al. [159] presented Name Server Switching Footprint (NSSF), a feature modality and a system that analyzes and detect the domain names with suspicious name server switching behavior. Additionally, in the same work, they have designed a time series prediction model to predict the number of name servers that a domain is likely to interact with.

Due to the critical role of the authoritative name servers in DNS infrastructure, they have been analyzed from a security point of view as well. For example, Vissers et al. [72] have investigated the security practices of authoritative name servers, and observed that a large number of domain names (≈1.28M) are under the risk of DoS attacks due to weak security practices of few (≈7,214) authoritative name servers. They also suggested employment of DNSSEC as a solution to enhance the security of DNS. Furthermore, Chung et al. [86] have conducted a measurement study on deployment and management of DNSSEC in both authoritative name servers and resolvers. Their results showed that more than 30% of signed domains failed to upload DS records, and some authoritative name servers failed to upload DS records for nearly all of their domains. Such practices, make DNSSEC ineffective and provide no practical security.

5.1.2. Recursive DNS server

A recursive DNS server is responsible to query other name servers to resolve a domain name. While recursive DNS resolver is significantly important in the function of DNS, miscreants have exploited it to launch various attacks, e.g., DNS amplification attack, cache poisoning, etc. Accordingly, researchers have explored various aspects of recursive DNS resolvers including, caching performance, DNS amplification attack, malware infection, etc. [155, 27, 21, 13, 22]. Chen et al. [21] have investigated the impact of disposable domains on the caching behavior of the DNS resolvers. Their analysis showed that as the prevalence of the disposable domains increases, it is likely that the DNS cache begins to fill up with RRset, that are unlikely to be reused. Hao and Wang [22] have studied the negative impact of one-time-use domain names on the performance of DNS caching. They argue that removal or not insertion of such RRset into the cache can prevent a waste of the DNS cache resources.

Shang et al. [156] explored the relationships across domain names to improve the cache hit rate of local DNS servers. In addition, Chen et al. [155] have presented a dynamic lease scheme that keeps track of local DNS name servers to provide stronger cache consistency and more reliable services. Perdisci et al. [19] have proposed a novel method based on WSEC DNS to mitigate recursive DNS resolvers cache poisoning attack.

Ager et al. [101] explored the impact of DNS resolvers latency and the DNS cache contents on the performance of local DNS and open DNS resolvers, e.g., GoogleDNS [160] and OpenDNS [161]. They further observed that unlike third-party DNS resolvers, local DNS resolvers redirect clients towards the content available within the ISP. Kuhrer et al. [36] conducted a large-scale study based on empirical data that is collected over one year, to investigate the landscape of DNS resolvers. Their analysis revealed that millions of these DNS resolvers deliberately manipulate resolutions for malicious purposes.

Verma et al. [13] have utilized query rate sharing property of DNS resolvers to build a DNS DDoS mitigation system that calculates global DNS query rate to make mitigation decisions locally. Moreover, Ballani et al. [7] have analyzed the caching behavior of DNS resolvers to defend against DNS DoS attacks. Discussion. Although top level DNS servers perform well, the quality of service is unfair around the globe [158]. Moreover, it is noteworthy that the vulnerability of a small number of authoritative name servers, potentially affects a large number of domains under their apex [72]. Authoritative name servers
are widely exploitable through outdated WHOIS email records of name server domains. Moreover, few authoritative name servers are responsible for incorrect deployment of DNSSEC in large number of domain names[86].

5.2. Open DNS Resolver

Contrary to the recursive DNS resolvers, that perform recursion for the internal clients only, an open DNS resolver resolves recursive DNS lookups for anyone over the Internet. As such, an open DNS resolvers is vulnerable to well known attacks, e.g., DDoS on behalf of attackers. An overview of recursive DNS resolvers and open DNS resolvers is shown in Figure 16 and Figure 17 respectively. To this end, extensive research efforts have been devoted to study the impact of open resolvers on the performance and security of DNS infrastructure [101, 17, 77, 36, 23, 37, 19, 93, 35, 13].

Based on the aforementioned properties of open resolvers, they can be exploited by the attackers to conduct a series of attacks. Significant research has been conducted to understand the DNS resolvers landscape and its associated threats.

5.2.1. Measurement on the DNS Protocol

Prior work on the analysis of DNS resolvers is mostly focused on a small subset of all resolvers. Therefore, it is unclear if the observations can be generalized on a wider scale to all the resolvers around the globe. For instance, Sisson [162] analyzed open resolvers based on sampled scans that repeatedly query the same set of resolvers. Therefore, their work was limited to only a small fraction of all the open resolvers. Antonakakis et al. [17] have analyzed a geographically diverse set of 300,000 open resolvers to measure the integrity of their responses. They observed that in general, the attackers point victims to rogue IP addresses. Furthermore, Jiang et al. [27] analyzed the caching behavior of resolvers and identified an attack vector in DNS software that allows the extension of domains caching even after their removal from the upper DNS hierarchy.

Schomp et al. [163] randomly probed the IPv4 address space to enumerate DNS resolvers and distinguish between recursive DNS resolvers and DNS proxies. They also performed an in-depth analysis on the caching behavior of the resolvers. Gao et al. [51] have analyzed a large set of DNS query-response pairs collected from over 600 recursive DNS resolvers. They observed that despite a great variation in the characteristics of the DNS traffic across networks, the behavior of resolvers within an organization remains similar.

5.2.2. Internet-Wide Scanning

Durumeric et al. [169] proposed a high-speed application to run Internet-wide scans called ZMap. ZMap is an open-source network scanner designed to perform Internet-wide scans with capability of surveying the entire IPv4 address space within 45 minutes. The Open Resolver Project [164, 165] actively investigates DNS servers world-wide and provides open resolver statistics on the web. The open resolver statistics are available online from March 2013 to January 2017, after which the scan was discontinued. Shadowserver [166] is another organization that conducts surveys related to the Internet security including active measurements of open resolvers, with up-to-date scans. Furthermore, Takano et al. [168] performed measurements based on responses for Internet-wide DNS software version requests. They have focused on DNS server software and their distribution in each regional Internet registry.

5.2.3. Threats

Antonakakis et al. [17] noticed that in general, the attackers point victims to rogue IP addresses through open DNS resolvers. Kuhrer et al. [36] have analyzed the threats in open resolvers from two perspectives. Firstly, they scanned the DNS resolvers’ landscape for changes in the course of time and categorized the resolvers based on their device type and the software version. Secondly, they measured the response authenticity of the resolvers from the users’ point-of-view to find that a large number of resolvers intentionally manipulate DNS resolutions for malicious activities.

Schomp et al. [35] have measured the vulnerability of the user-side DNS infrastructure to record injection threats. They have found that many open DNS resolvers, that are vulnerable to record injection attacks, are used as an attack vector to target shared DNS infrastructure. Verma et al. [13] presented a system to mitigate Amplified DNS DDoS (ADD) attacks, that relies on the fact that DNS resolvers share local DNS query rates that can be used to calculate the total query rates.

Ballani et al. [7] have presented a defense scheme called stale cache, that uses caching behavior to prevent DNS DoS attacks. They have analyzed DNS traces under multiple DoS attack strategies and have found that the cost-benefit ratio of the proposed method, favors the deployment of stale cache. Furthermore, Herrmann et al. [77] have studied third party DNS
resolvers and consequent threats to online privacy. They proposed a new privacy-preserving name resolution service that relies on the encapsulation of encrypted messages in standards-compliant DNS messages. Hendriks et al. [170] presented an active measurement system to find the list of open DNS resolvers on IPv6 in the wild, which could be potentially exploited in a DDoS attack. Moreover, Schomp et al. [93] have studied the vulnerabilities of shared DNS resolvers and proposed an unorthodox method to tackle such threats.

**Discussion.** DNS resolvers can be easily discovered and used by the attackers for malicious activities, such as DoS attacks. Moreover, buggy implementations and large DNSSEC response filtering can lead to a high query latency [158]. Although extensive research has been conducted on open DNS resolvers, their evolution and distribution still require a clear demonstration. Some open questions that are worth addressing in this regard are: 1. How is the distribution of open DNS resolvers in popular TLDs? 2. Are open resolvers consistent in answering various clients for the same query type?

5.3. **Hosting services**

The task for remedying compromised web resources is shared between hosting providers and webmasters. Shared hosting providers retain more control over configurations, which explains their association with the widespread abuse.

In [93], Adrichem et al. have conducted measurements to identify and categorize potential causes of the DNSSEC misconfigurations based on the reachability of a zone’s network. They have analyzed domains in six zones, including .bg, .br, .co, .com, .nl and .se, and noticed that a small number of hosting providers are responsible for DNSSEC misconfigured domains. They concluded that the misconfigured domains are at the risk of being unreachable from DNSSEC-aware resolver. DNSSEC allows clients to verify the integrity and authenticity of DNS records. However, DNSSEC has witnessed a low deployment rate with only 0.6% of .com, 0.8% of .net, and 1% of .org properly signed domains [86]. Chung et al. [86] have attempted to identify the impact of registrars on deployment of DNSSEC, since the registrars usually serve as DNS operators to their customers. A DNS operator can be a registrar, an owner, or a third-party operator, responsible for maintaining DNSKEY and RRSIG records. Based on the registrars’ policy of uploading records, DNSSEC can be fully deployed (with DS record), or partially deployed (without DS record).

Hao et al. [154] have conducted a measurement study to understand the authoritative DNS servers’ deployment patterns in modern web services. They have explored several characteristics DNS servers including, performance, availability, and lifecycle of servers. They have heuristically analyzed the Alexa’s top 1-milion list to identify the authoritative DNS deployment patterns in web domains. Their results show that most of the emerging popular social websites host the authoritative DNSes in upstream services that provide performance advantages. Furthermore, they observed that backup and redundant deployment in hybrid patterns provide availability; revealing the growth of cloud providing DNS hosting services. Herzberg et al. [9] have studied DNS-amplification DoS attacks, and have proposed a defense system that is compatible with common DNS servers configurations and DNSSEC. The have shown the efficiency and high performance of the presented DNS-authentication method in preventing DNS-based amplification DoS attacks. They have also predicted the adoption of their design, based on game theory algorithms, concluding that their design will sufficiently reduce DNS amplification DoS attacks. Additionally, their proposed method can be deployed as a cloud-based service, that can reduced cost and upgrade defenses for DNS servers.

**Discussion.** Cloud services are increasingly popular due to their affordability [154]. However, they are also subject to DoS attacks, where clients are at risk due to poorly written programs. Clients of shared hosting providers share IP address with other clients, which means that blocking one IP results in blocking all the users. Therefore, the security of cloud services is critical and should be well established.

5.4. **Client**

Although, much knowledge about DNS infrastructure has been derived from aggregate population of clients’ behavior
analysis, researchers have also emphasized on individual client’s interaction with DNS ecosystem. For example, Schomp et al. have analyzed the behavior of individual DNS users in order to develop an analytical model of their interaction with the DNS ecosystem. They observed that different types of users behave uniquely with DNS. In addition, they have demonstrated that the combination of the Weibull and Pareto can successfully model the process of client query arrival. Finally, they observed the existence of a fairly stable and unique working set of names for each client.

Pan et al. have proposed a novel client classification method based on client query entropy and global recursive DNS service architecture. By monitoring the query frequencies of the clients, they have validated the effectiveness of the proposed method on busy and long-tailed clients. They have found that 2.32% clients can cover the most important web spiders, recursive servers, and well-known internet services. Greschbach et al. have investigated the impact of DNS traffic on Tor clients’ vulnerability to correlation attacks and how DNS lookups can be utilized for information stealth.

Kuhrer et al. have highlighted the vulnerabilities in DNS resolvers by analyzing the changes in their landscape, and the response authenticity of the resolvers from the users’ perspective. They observed that a large number of resolvers intentionally manipulate resolutions for malicious activities.

Khan et al. have measured the harm caused to the users by quantitatively measuring the time wasted by typosquatting. They have introduced a new metric that offers several advantages such as empirical quantification of harm to the clients, identification of typosquatting domain names, and proportion of different typosquatting perpetrators. They have analyzed a large scale DNS dataset (active and passive), and noticed that the typosquatting events affect both the users and the websites. Their analysis confirms that typosquatting increases both the time taken by the client to find the intended website, and the latency between a typo and its correction.

Luo et al. have analyzed ISP network traces to understand various approaches of the attackers towards the exploitation of DNS for malicious activities. They have taken a variety of syntactic as well as temporal patterns into account to propose a method that identifies different clusters of malicious domain names that lead to DNS failures. Based on their evolutionary learning framework, less suspicious clusters were removed while highly suspicious cases were preserved. Their proposed framework analyzes DNS failures on per-client basis and in practice, they have used their framework on a large-scale ISP network trace to find that over 97% of the users with suspicious DNS activities can be detected with 81% precision.

Discussion. Clients are usually the main target in cyber crimes, and they suffer from a wide variety of attacks including phishing attacks, DNS manipulation, cybersquatting, etc. Existing research in the literature has analyzed the impact of threats on the clients, however, it remains an open challenge to extend the literature for user-intention-based anomaly detection method to identify anomalous DNS traffic.

6. Conclusion

In this paper, a comprehensive study have been conducted to review the ecosystem of DNS from various points of view. Therefore, a large number of peer reviewed and recently published research works have been surveyed and then potential vulnerabilities of DNS security and corresponding countermeasures have been summarized. In addition, we have looked into the research works from two different point of view, namely data analysis methods and DNS entities landscape. In DNS research methods, we have looked into the literature to understand the limitations as well as strengths of utilized data analysis methods which would offer valuable guideline to analyze the existing challenges in DNS ecosystem. The second, DNS entities scope, looks into the role of different entities in DNS infrastructure and points out common vulnerabilities and limitations of existing components of the DNS ecology. Furthermore, in each sub-section we have discussed the challenges and pointed out open research directions.

7. Acknowledgement

This work is supported in part by National Research Foundation of South Korea under grant NRF-2016K1A1A2912757.

References

References

[1] Futai Zou, Siyu Zhang, Bei Pei, Li Pan, Linsen Li, and Jianhua Li. Survey on domain name system security. In Proceedings of the IEEE First International Conference on Data Science in Cyberspace DSC, pages 602–607, 2016.

[2] Ah Reum Kang, Jeffrey Spaulding, and Aziz Mohaisen. Domain name system security and privacy: Old problems and new challenges. CoRR, abs/1606.07080, 2016.

[3] Sadegh Torabi, Amine Boukhitouta, Chadi Ass, and Mourad Debbi. Detecting internet abuse by analyzing passive dns traffic: A survey of implemented systems. IEEE Communications Surveys & Tutorials, 2018.

[4] Yury Zhauniarovich, Isha Khalil, Ting Yu, and Marc Dacier. A survey on malicious domains detection through dns data analysis. ACM Computing Surveys, 51(4), July 2018.

[5] Jeffrey Spaulding, Shambhu J. Upadhyaya, and Aziz Mohaisen. The landscape of domain name typosquatting: Techniques and countermeasures. In Proceedings of the 11th International Conference on Availability, Reliability and Security ARES, pages 284–289, 2016.

[6] Ahmad Ariff Azizuddin, Mohd Atan, Megat Norulazmi, Megat Mohamed Noor, Shadil Akimi, and Zainal Abidin. DNS amplification attack detection and mitigation via sflow with security-centric SDN. In Proceedings of the 11th International Conference on Ubiquitous Information Management and Communication IMCOM, page 3, 2017.

[7] Hitesh Ballani and Paul Francis. A simple approach to dns dos mitigation. Microsoft, 2016.

[8] Soyoung Kim, Sora Lee, Gseunhwan Cho, Muhammad Ejaz Ahmed, Jaeheon (Paul) Jeong, and Hyounsikh Kim. Preventing DNS amplification attacks using the history of DNS queries with SDN. In Proceedings of the 22nd European Symposium on Research in Computer Security (ESORICS), pages 135–152, Oslo, Norway, September 2017.
[9] Amir Herzberg and Haya Shulman. DNS authentication as a service: preventing amplification attacks. In Proceedings of the 30th Annual Computer Security Applications Conference ACSAC, pages 356–365, 2014.

[10] Douglas C. MacFarland, Craig A. Shue, and Andrew J. Kalafut. Characterizing optimal DNS amplification attacks and effective mitigation. In Proceedings of the 16th International Conference on Passive and Active Measurement PAM, pages 15–27, New York, NY, March 2015.

[11] Roberto Perdisci, Igino Corona, and Giorgio Giacinto. Early detection of malicious flux networks via large-scale passive DNS traffic analysis. IEEE Transactions on Dependable and Secure Computing, 9(5):714–726, 2012.

[12] Dmihu Traong and Guang Cheng. Detecting domain-flux botnet based on DNS traffic features in managed network. Security and Communication Networks, 9(14):2338–2347, 2016.

[13] Saurabh Verma, Ali Hameeh, Jun Ho Huik, Henrik Holm, Siva Raj Rajagopalan, Maciej Korczynski, and Nina H. Fefferman. Stopping amplified DNS ddos attacks through distributed query rate sharing. In Proceedings of the 11th International Conference on Availability, Reliability and Security ARES, pages 69–78, 2016.

[14] Jing Zheng, Qi Li, Guofei Gu, Jiahao Cao, and David KY Yau. Realtime ddos attacks with cots sdn switches via adaptive correlation analysis. IEEE Transactions on Information Forensics and Security, 13(7):1838–1853, 2018.

[15] Nexusguard Limited. Ddos threats report 2018 q1.

[16] Haya Shulman and Michael Waidner. Towards forensic analysis of attacks with DNSSEC. In Proceedings of the 35th IEEE Symposium on Security and Privacy (Oakland), pages 69–76, San Jose, CA, May 2014.

[17] Elisa Bertino and Nayeem Islam. Botnets and internet of things security. Computer, 50(2):76–79, 2017.

[18] Constantinos Kollis, Georgios Kambourakis, Angelos Stavrou, and Jefrey Voas. Ddos in the it: Mirai and other botnets. Computer, 50(7):80–84, 2017.

[19] Elisa Bertino and Nayeem Islam. Botnets and internet of things security. Computer, 50(2):76–79, 2017.

[20] Nicholas Weaver, Christian Kreibich, Boris Nechaev, and Vern Paxson. Implications of netalyzers dns measurements. In Proceedings of the First Workshop on Securing and Trusting Internet Names (SATIN). 2011.

[21] Kyle Schomp, Tom Callahan, Michael Robinovich, and Mark Allman. Assessing DNS vulnerability to record injection. In Proceedings of the 15th International Conference on Passive and Active Measurement PAM, pages 214–223, Los Angeles, CA, March 2014.

[22] Marc Kuehner, Thomas Hupperich, Jonas Bushart, Christian Rossov, and Thorsten Holz. Going wild: Large-scale classification of open DNS resolvers. In Proceedings of the 2015 Internet Measurement Conference (IMC), pages 355–368, Toyoko, Japan, October 2015.

[23] Paul Pearce, Ben Jones, Frank Li, Raya Ensafi, Nick Feamster, Nick Weaver, and Vern Paxson. Global measurement of DNS manipulations. In Proceedings of the 26th USENIX Security Symposium (Security), pages 307–323, Vancouver, BC, Canada, August 2017.

[24] Martino Trevisan, Idilio Drago, Marco Mellia, and Maurizio M. Munafò. Automatic detection of DNS manipulations. In Proceedings of the IEEE International Conference on Big Data, pages 4010–4017, 2017.

[25] Leyla Bilge, Engin Kirda, Christopher Kruegel, and Marco Balduzzi. EXPOSURE: finding malicious domains using passive DNS analysis. In Proceedings of the 2015 Annual Network and Distributed System Security Symposium (NDSS), San Diego, CA, February 2011.

[26] David Dagon, Niels Provos, Christopher P. Lee, and Wenke Lee. Combosquatting abuse. In Proceedings of the 24th ACM Conference on Computer and Communications Security (CCS), pages 182–186, Dallas, TX, October–November 2017.

[27] Debra L. Cook, Vijay K. Gurbani, and Michael Daniluk. Phishwish: A main name detection based on machine learning. In Proceedings of the 12th International Conference on Financial Cryptography and Data Security FC, pages 182–186, 2008.

[28] Eric Medvet, Engin Kirda, and Christopher Kruegel. Visual-similarity-based phishing detection. In Proceedings of the 4th International ICST Conference on Security and Privacy in Communication Networks SECURECOMM, page 22, 2008.

[29] Zack Whittaker. Mirai botnet attackers are trying to knock an entire country offline. Nov 2016.

[30] Danny Palmer. Mirai botnet adds three new attacks to target iot devices, May 2018.

[31] Roberto Perdisci, Igino Corona, and Giorgio Giacinto. Early detection of malicious flux networks via large-scale passive DNS traffic analysis. IEEE Transactions on Dependable and Secure Computing, 9(5):714–726, 2012.

[32] Paul Pearce, Ben Jones, Frank Li, Raya Ensafi, Nick Feamster, Nick Weaver, and Vern Paxson. Global measurement of DNS manipulations. In Proceedings of the 26th USENIX Security Symposium (Security), pages 307–323, Vancouver, BC, Canada, August 2017.

[33] Elisa Bertino and Nayeem Islam. Botnets and internet of things security. Computer, 50(2):76–79, 2017.

[34] Constantinos Kollis, Georgios Kambourakis, Angelos Stavrou, and Jefrey Voas. Ddos in the it: Mirai and other botnets. Computer, 50(7):80–84, 2017.

[35] Elisa Bertino and Nayeem Islam. Botnets and internet of things security. Computer, 50(2):76–79, 2017.
Manos Antonakakis, Roberto Perdisci, David Dagon, Wenke Lee, and Nick Feamster. Building a dynamic reputation system for DNS. In Proceedings of the 19th USENIX Security Symposium (Security), pages 273–290, Washington, DC, August 2010. 2, 10

Manos Antonakakis, Roberto Perdisci, Wenke Lee, Nikolaos Vasiloglou, and David Dagon. Detecting malware domains at the upper DNS hierarchy. In Proceedings of the 20th USENIX Security Symposium (Security), San Francisco, CA, August 2011. 2, 10

Leyla Bilge, Sevil Sen, Davide Balzarotti, Engin Kirda, and Christopher Kruegel. Exposure: A passive DNS analysis service to detect and report malicious domains. ACM Transactions on Information and System, 3(4):1–14, 2014. 2, 15

Davide Canalli, Marco Costa, Giovanni Vigna, and Christopher Kruegel. Proceedings of the 20th international world wide web conference (www). In Proceedings of the 20th International World Wide Web Conference (WWW), pages 197–206, Hyderabad, India, April 2011. 2, 10

Mark Félegyházi, Christian Kreibich, and Vern Paxson. On the potential of proactive domain blacklisting. In Proceedings of the 3rd USENIX Workshop on Large-Scale Exploits and Emergent Threats, 2010. 2, 10

Hongyu Gao, Vinod Yegneswaran, Yan Chen, Phillip A. Porras, Shalini Ghosh, Jian Jiang, and Hai-Xin Duan. An empirical reexamination of global DNS behavior. In Proceedings of the 2013 ACM SIGCOMM, pages 267–278, Hong Kong, China, August 2013. 2, 10

Shuang Hao, Alex Kandchelian, Brad Miller, Vern Paxson, and Nick Feamster. PREDATOR: proactive recognition and elimination of domain abuse at time-of-registration. In Proceedings of the 23rd ACM Conference on Computer and Communications Security (CCS), pages 1568–1579, Vienna, Austria, October 2016. 2, 10

Shuang Hao, Matthew Thomas, Vern Paxson, Nick Feamster, Christian Kreibich, Chris Grier, and Scott Hollenbeck. Understanding the domain registration behavior of spammers. In Proceedings of the Internet Measurement Conference (IMC), pages 63–76, Barcelona, Spain, October 2013. 2, 10

Issa Khalil, Bei Guan, Mohamed Nabeel, and Ting Yu. Killing two birds with one stone: Malicious domain detection with high accuracy and coverage. In Proceedings of the 8th ACM Conference on Data and Application Security and Privacy, 2018. 2, 10

Pengkui Luo, Ruben Torres, Zhu-Li Zhang, Sabyasachi Saha, Sung-Ju Lee, Antonio Nucci, and Marco Mellia. Leveraging client-side DNS failure patterns to identify malicious behaviors. In Proceedings of the IEEE Conference on Communications and Network Security (CNS), pages 406–414, 2015. 2, 10

Justin Ma, Lawrence K. Saul, Stefan Savage, and Geoffrey M. Voelker. Identifying suspicious urls: an application of large-scale online learning. In Proceedings of the 25th International Conference on Machine Learning (ICML), pages 681–688, Montreal, Canada, June 2009. 2, 10

Shuang Hao, Nick Feamster, and Ramakant Pandrangi. Monitoring the initial DNS behavior of malicious domains. In Proceedings of the International Measurement Conference (IMC), pages 269–278, Berlin, Germany, November 2011. 2, 10

Daiping Liu, Zhou Li, Kun Du, Haining Wang, Baojun Liu, and Hai-Xin Duan. Don’t let one rotten apple spoil the whole barrel: Towards automated detection of shadowed domains. In Proceedings of the 24th ACM Conference on Computer and Communications Security (CCS), pages 537–552, Dallas, TX, October–November 2017. 2, 10

Sandeep Yadav, Ashwath Kumar Krishna Reddy, A. L. Narasimha Reddy, and Supranamaya Ranjan. Detecting algorithmically generated malicious domain names. In Proceedings of the Internet Measurement Conference (IMC), pages 48–61, Melbourne, Australia, November 2010. 2, 10

Manos Antonakakis, Roberto Perdisci, Yacin Nadj, Nikolaos Vasiloglou, Saeed Abutahhan, Wenke Lee, and David Dagon. From throw-away traffic to bots: Detecting the rise of dga-based malware. In Proceedings of the 21st USENIX Security Symposium (Security), volume 12, Bellevue, WA, August 2012. 2, 10

Thom Barabosch, Andre Wichmann, Felix Leder, and Elmar Gerhards-Padilla. Automatic extraction of domain name generation algorithms from current malware. In Proceedings of the NATO Symposium on Information Assurance and Cyber Defense, 2012. 2, 10

Jeffrey Spaulding, Jemam Park, Jongheon Kim, and Aziz Mohaisen. Proactive detection of algorithmically generated malicious domains. In Proceedings of the International Conference on Information Networking, ICIN, pages 21–24, 2016. 2, 10

Yi-Min Wang, Doug Beck, Jeffrey Wang, Chad Verbowski, and Brad Daniels. Strider typo-patrol: Discovery and analysis of systematic typo-squatting. In Proceedings of the 2nd Workshop on Steps to Reducing Unwanted Traffic on the Internet, SRUTI’06, 2006. 2, 10

Anirban Banerjee, Dhiman Barman, Michalis Faloutsos, and Laxmi N. Bhuyan. Cyber-fraud is one typo away. In Proceedings of the 27th IEEE International Conference on Computer Communications, Joint Conference of the IEEE Computer and Communications Societies, INFOCOM, pages 1939–1947, 2006. 2, 10

Pieter Agten, Wouter Joosen, Frank Piessens, and Nick Nikiforakis. Seven months’ worth of mistakes: A longitudinal study of typosquatting abuse. In Proceedings of the 2015 Annual Network and Distributed System Security Symposium (NDSS), San Diego, CA, February 2015. 2, 10

Janos Szurdi, Balazs Kocso, Gabor Csh, Jonathan Spring, Mark Félegyházi, and Chris Kanich. The long “taile” of typosquatting domain names. In Proceedings of the 23rd USENIX Security Symposium (Security), pages 191–206, San Diego, CA, August 2014. 2, 10

Mohammad Taha Khan, Xiang Huo, Zhou Li, and Chris Kanich. Every second counts: Quantifying the negative externalities of cybercrime via typosquatting. In Proceedings of the 36th IEEE Symposium on Security and Privacy (Oakland), pages 135–150, San Jose, CA, May 2015. 2, 10

Jeffrey Spaulding, DaHun Nyang, and Aziz Mohaisen. Understanding the effectiveness of typosquatting techniques. In Proceedings of the fifth ACM/IEEE Workshop on Hot Topics in Web Systems and Technologies, HotWeb, pages 9:1–9:8, 2017. 2, 10

Jeffrey Spaulding, Shambhu J. Upadhyaya, and Aziz Mohaisen. You’ve been tricked! A user study of the effectiveness of typosquatting techniques. In Proceedings of the 37st International Conference on Distributed Computing Systems (ICDCS), pages 2593–2596, 2017. 2, 10

Bitsquatting: Dns hijacking without exploitation. 2, 13

Nick Nikiforakis, Steven Van Acker, Wannes Meert, Lieven Desmet, Frank Piessens, and Wouter Joosen. Bitsquatting: exploiting bit-flips for fun, or profit? In Proceedings of the 22nd International World Wide Web Conference (WWW), pages 989–998, 2013. 2, 10

Thomas Vissers, Timothy Barron, Tom van Goethem, Wouter Joosen, and Nick Nikiforakis. The wolf of name street: Hijacking domains through their nameservers. In Proceedings of the 24th ACM Conference on Computer and Communications Security (CCS), pages 957–970, Dallas, TX, October–November 2017. 2, 10

Nick Nikiforakis, Marco Balduzzi, Lieven Desmet, Frank Piessens, and Wouter Joosen. Soundsquatting: Uncovering the use of homophones in domain squatting. In Proceedings of the 17th Information Security Conference (ISC), pages 291–308, Hong Kong, October 2014. 2, 10

Sumayyah A. Alrwais, Kan Yuan, Eihal Alowaisheq, Zhou Li, and Xiaofeng Wang. Understanding the dark side of domain parking. In Proceedings of the 23rd USENIX Security Symposium (Security), pages 207–222, San Diego, CA, August 2014. 2, 10

Nicholas Weaver, Christian Kreibich, and Vern Paxson. Redirecting DNS for ads and profit. In Proceedings of the USENIX Workshop on Free and Open Communications on the Internet FOCS, 2011. 2, 10

Thomas Vissers, Wouter Joosen, and Nick Nikiforakis. Parking sensors: Analyzing and detecting parked domains. In Proceedings of the 2015 Annual Network and Distributed System Security Symposium (NDSS), San Diego, CA, February 2015. 2, 10

Dominik Herrmann, Karl-Peter Fuchs, Jens Lindemann, and Hannes Feuderrath. Encdns: A lightweight privacy-preserving name resolution service. In Proceedings of the 19th European Symposium on Research in Computer Security (ESORICS), pages 37–55, Wroclaw, Poland, September 2014. 2, 10

Yanbin Lu and Gene Tsudik. Towards plugging privacy leaks in the domain name system. In Proceedings of the 10th IEEE International
Conference on Peer-to-Peer Computing P2P, pages 1–10, 2010.

Aziz Mohaisen, Zhongshu Gu, and Kui Ren. Privacy implications of DNSSEC look-aside validation. In Proceedings of the 37th International Conference on Distributed Computing Systems (ICDCS), pages 2434–2443, 2017.

Sumayah A. Alrwais, Xiaojing Liao, Xianghang Liu, Meng Wang, Xiaofeng Wang, Feng Qian, Raheem A. Beyah, and Damon McCoy. Under the shadow of sunshine: Understanding and detecting bulletproof hosting on legitimate service provider networks. In Proceedings of the 38th IEEE Symposium on Security and Privacy (Oakland), pages 805–823, San Jose, CA, May 2017.

Thomas Callahan, Mark Allman, and Michael Rabinovich. On modern DNS behavior and properties. Computer Communication Review, 43(3):7–15, 2013.

Samaneh Tajalizadehkhoo, Tom van Goethem, Maciej Korczyński, Arman Noroozian, Rainer Böhme, Tyler Moore, Wouter Joosen, and Michel van Eeten. Herding vulnerable cats: A statistical approach to disentangle joint responsibility for web security in shared hosting. In Proceedings of the 24th ACM Conference on Computer and Communications Security (CCS), pages 553–567, Dallas, TX, October–November 2017.

Athanasios Kontouras, Panagiotis Kintis, Chaz Lever, Yi zheng Chen, Yacin Nadji, David Dagon, Manos Antonakakis, and Rodney Joffe. Enabling network security through active DNS datasets. In Proceedings of the 19th International Symposium on Research in Attacks, Intrusions and Defenses (RAID), pages 188–208, Evry, France, September 2016.

Zhou Li, Sumayah A. Alrwais, Yinglian Xie, Fang Yu, and Xiaofeng Wang. Finding the linchpins of the dark web: a study on topologically dedicated hosts on malicious web infrastructures. In Proceedings of the 34th IEEE Symposium on Security and Privacy (Oakland), pages 112–126, San Francisco, CA, May 2013.

Roland van Rijswijk-Deij, MattiuS Jonker, Anna Sperotto, and Aiko Pras. A high-performance, scalable infrastructure for large-scale active dns measurements. IEEE journal on selected areas in communications, 34(6):1877–1888, 2016.

Taejoong Chung, Roland van Rijswijk-Deij, Balakrishnan Chandrasekaran, David R. Choffnes, Dave Levin, Bruce M. Maggs, Alon Mislove, and Christo Wilson. A longitudinal, end-to-end view of the DNSSEC ecosystem. In Proceedings of the 26th USENIX Security Symposium (USENIX Security), pages 1307–1322, Vancouver, BC, Canada, August 2017.

Marwan Rashwan and Rekso Heckel. Prediction of the domain name system (dns) quality attributes. In Proceedings of the Symposium on Applied Computers, pages 578–585, 2017.

Babak Rahbarinia, Roberto Per duschi, and Manos Antonakakis. Seguaggio: Efficient behavior-based tracking of malware-control domains in large ISP networks. In Proceedings of the 45th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN), pages 403–414, 2015.

Haya Shulman and Michael Waidner. Towards security of internet naming infrastructure. In Proceedings of the 20th European Symposium on Research in Computer Security (ESORICS), pages 3–22, Vienna, Austria, September 2015.

Filip Hock and Peter Kotrén. Design, implementation and monitoring of the firewall system for a dns server protection. In Proceedings of the International Conference on Emerging eLearning Technologies and Applications ICTEA, pages 91–96, 2016.

Rosa Romero Gómez, Yacin Nadji, and Manos Antonakakis. Towards designing effective visualizations for dns-based network threat analysis. In Proceedings of the IEEE Symposium on Visualization for Cyber Security, pages 1–8, 2017.

Collin Jackson, Adam Barth, Andrew Bortz, Weidong Shao, and Dan Boneh. Protecting browsers from DNS rebinding attacks. ACM Transactions on the Web, 9(4):1–26, 2009.

Kyle Schomp, Mark Allman, and Michael Rabinovich. DNS resolvers considered harmful. In Proceedings of the 13th ACM Workshop on Hot Topics in Networks HotNets, pages 16:1–16:7, 2014.

David Dagon, Manos Antonakakis, Paul Vixie, Tatjana Jimm, and Wenke Lee. Increased DNS forgery resistance through 6x20-bit encod-
ciency. The annals of mathematical statistics, 22(1):79–86, 1951.

[10] Alasdair King. Thunder screenreader.

[11] Welcome to orca.

[12] Talk to your internet.

[13] Fangming Zhao, Yoshiaki Hori, and Kouichi Sakurai. Two-servers PIR based DNS query scheme with privacy-preserving. In Proceedings of the 2007 International Conference on Multimedia and Ubiquitous Engineering MUE, pages 952–957, 2007.

[14] Fangming Zhao, Yoshiaki Hori, and Kouichi Sakurai. Analysis of privacy disclosure in DNS query. In Proceedings of the 2007 International Conference on Intelligent Pervasive Computing, IPC, pages 299–302, Jeju City, South Korea, October 2007.

[15] Liang Zhu, Zi Hu, John S. Heidemann, Duane Wessels, Allison Mankin, and Nikita Somaiya. Connection-oriented DNS to improve privacy and security. In Proceedings of the 36th IEEE Symposium on Security and Privacy (Oakland), pages 171–186, San Jose, CA, May 2015.

[16] Meeto and Nikita Somaiya. Control over DNS: Understanding the security threats of dangling DNS records. In Proceedings of the 3rd USENIX Workshop on Large-Scale Exploits and Emergent Threats LEET, 2010.

[17] Sergio Castillo-Perez and Joaquín García-Aflao. Evaluation of two privacy-preserving protocols for the DNS. In Proceedings of the 6th International Conference on Information Technology: New Generations ITING, pages 411–416, 2009.

[18] Vern Paxson, Mihai Chiroscordescu, Mohin Javed, Josyula R. Rao, Reiner Sailer, Douglas Lee Schales, Marc Ph. Steecklin, Kurt Thomas, Wietse Venema, and Nicholas Weaver. Practical comprehensive bounds on surreptitious communication over DNS. In Proceedings of the 22th USENIX Security Symposium (Security), pages 17–32, Washington, DC, August 2013.

[19] Daiping Liu, Shuq Hao, and Haining Wang. All your DNS records point to us: Understanding the security threats of dangling DNS records. In Proceedings of the 23rd ACM Conference on Computer and Communications Security (CCS), pages 1414–1425, Vienna, Austria, October 2016.

[20] Aziz Mohaisen and Kui Ren. Leakage of onion at the DNS root: Measurements, causes, and countermeasures. IEEE/ACM Transactions on Networking, 25(5):3059–3072, 2017.

[21] Aziz Mohaisen, Ah Reum Kang, and Kui Ren. Does query blocking improve DNS privacy? - quantifying privacy under partial blocking deployment. In Proceedings of the 17th International Workshop on Information Security Applications, WISA, pages 1–14, Jeju Island, Korea, August 2016.

[22] Haya Shulman. Pretty bad privacy: Pitfalls of DNS encryption. In Proceedings of the 13th Workshop on Privacy in the Electronic Society, WPES, pages 191–200, 2014.

[23] DNS over dtls (dnsod). 13

[24] The "onion" special-use domain name. 14

[25] Srinivas Krishnan and Fabian Monrose. DNS prefetching and its privacy implications: When good things go bad. In Proceedings of the 3rd USENIX Workshop on Large-Scale Exploits and Emergent Threats LEET, 2010.

[26] Alasdair King. Thunder screenreader.

[27] Vern Paxson, Mihai Chiroscordescu, Mohin Javed, Josyula R. Rao, Reiner Sailer, Douglas Lee Schales, Marc Ph. Steecklin, Kurt Thomas, Wietse Venema, and Nicholas Weaver. Practical comprehensive bounds on surreptitious communication over DNS. In Proceedings of the 22th USENIX Security Symposium (Security), pages 17–32, Washington, DC, August 2013.

[28] Daiping Liu, Shuq Hao, and Haining Wang. All your DNS records point to us: Understanding the security threats of dangling DNS records. In Proceedings of the 23rd ACM Conference on Computer and Communications Security (CCS), pages 1414–1425, Vienna, Austria, October 2016.

[29] Aziz Mohaisen and Kui Ren. Leakage of onion at the DNS root: Measurements, causes, and countermeasures. IEEE/ACM Transactions on Networking, 25(5):3059–3072, 2017.

[30] Aziz Mohaisen, Ah Reum Kang, and Kui Ren. Does query blocking improve DNS privacy? - quantifying privacy under partial blocking deployment. In Proceedings of the 17th International Workshop on Information Security Applications, WISA, pages 1–14, Jeju Island, Korea, August 2016.

[31] Haya Shulman. Pretty bad privacy: Pitfalls of DNS encryption. In Proceedings of the 13th Workshop on Privacy in the Electronic Society, WPES, pages 191–200, 2014.

[32] DNS over dtls (dnsod). 13

[33] The "onion" special-use domain name. 14

[34] Srinivas Krishnan and Fabian Monrose. DNS prefetching and its privacy implications: When good things go bad. In Proceedings of the 3rd USENIX Workshop on Large-Scale Exploits and Emergent Threats LEET, 2010.

[35] Alasdair King. Thunder screenreader.

[36] Vern Paxson, Mihai Chiroscordescu, Mohin Javed, Josyula R. Rao, Reiner Sailer, Douglas Lee Schales, Marc Ph. Steecklin, Kurt Thomas, Wietse Venema, and Nicholas Weaver. Practical comprehensive bounds on surreptitious communication over DNS. In Proceedings of the 22th USENIX Security Symposium (Security), pages 17–32, Washington, DC, August 2013.

[37] Daiping Liu, Shuq Hao, and Haining Wang. All your DNS records point to us: Understanding the security threats of dangling DNS records. In Proceedings of the 23rd ACM Conference on Computer and Communications Security (CCS), pages 1414–1425, Vienna, Austria, October 2016.

[38] Aziz Mohaisen and Kui Ren. Leakage of onion at the DNS root: Measurements, causes, and countermeasures. IEEE/ACM Transactions on Networking, 25(5):3059–3072, 2017.

[39] Aziz Mohaisen, Ah Reum Kang, and Kui Ren. Does query blocking improve DNS privacy? - quantifying privacy under partial blocking deployment. In Proceedings of the 17th International Workshop on Information Security Applications, WISA, pages 1–14, Jeju Island, Korea, August 2016.

[40] Haya Shulman. Pretty bad privacy: Pitfalls of DNS encryption. In Proceedings of the 13th Workshop on Privacy in the Electronic Society, WPES, pages 191–200, 2014.

[41] DNS over dtls (dnsod). 13

[42] The "onion" special-use domain name. 14

[43] Srinivas Krishnan and Fabian Monrose. DNS prefetching and its privacy implications: When good things go bad. In Proceedings of the 3rd USENIX Workshop on Large-Scale Exploits and Emergent Threats LEET, 2010.

[44] Alasdair King. Thunder screenreader.

[45] Vern Paxson, Mihai Chiroscordescu, Mohin Javed, Josyula R. Rao, Reiner Sailer, Douglas Lee Schales, Marc Ph. Steecklin, Kurt Thomas, Wietse Venema, and Nicholas Weaver. Practical comprehensive bounds on surreptitious communication over DNS. In Proceedings of the 22th USENIX Security Symposium (Security), pages 17–32, Washington, DC, August 2013.

[46] Daiping Liu, Shuq Hao, and Haining Wang. All your DNS records point to us: Understanding the security threats of dangling DNS records. In Proceedings of the 23rd ACM Conference on Computer and Communications Security (CCS), pages 1414–1425, Vienna, Austria, October 2016.

[47] Aziz Mohaisen and Kui Ren. Leakage of onion at the DNS root: Measurements, causes, and countermeasures. IEEE/ACM Transactions on Networking, 25(5):3059–3072, 2017.

[48] Aziz Mohaisen, Ah Reum Kang, and Kui Ren. Does query blocking improve DNS privacy? - quantifying privacy under partial blocking deployment. In Proceedings of the 17th International Workshop on Information Security Applications, WISA, pages 1–14, Jeju Island, Korea, August 2016.

[49] Haya Shulman. Pretty bad privacy: Pitfalls of DNS encryption. In Proceedings of the 13th Workshop on Privacy in the Electronic Society, WPES, pages 191–200, 2014.

[50] DNS over dtls (dnsod). 13

[51] The "onion" special-use domain name. 14

[52] Srinivas Krishnan and Fabian Monrose. DNS prefetching and its privacy implications: When good things go bad. In Proceedings of the 3rd USENIX Workshop on Large-Scale Exploits and Emergent Threats LEET, 2010.

[53] Alasdair King. Thunder screenreader.

[54] Vern Paxson, Mihai Chiroscordescu, Mohin Javed, Josyula R. Rao, Reiner Sailer, Douglas Lee Schales, Marc Ph. Steecklin, Kurt Thomas, Wietse Venema, and Nicholas Weaver. Practical comprehensive bounds on surreptitious communication over DNS. In Proceedings of the 22th USENIX Security Symposium (Security), pages 17–32, Washington, DC, August 2013.

[55] Daiping Liu, Shuq Hao, and Haining Wang. All your DNS records point to us: Understanding the security threats of dangling DNS records. In Proceedings of the 23rd ACM Conference on Computer and Communications Security (CCS), pages 1414–1425, Vienna, Austria, October 2016.
Kyle Schomp, Tom Callahan, Michael Rabinovich, and Mark Allman. On measuring the client-side DNS infrastructure. In Proceedings of the Internet Measurement Conference (IMC), pages 77–90, Barcelona, Spain, October 2013.

Jared Mauch. Open resolver project. In Presentation, DNS-OARC Workshop, Dublin, May 12, 2013.

Open resolver project.

Shadowserver.

Duane Wessels and Aziz Mohaisen. Open resolvers in com/net resolution. Technical report, Domain Name System Operations, Analysis, and Research Center, DNS-OARC, 2014.

Yuki Takano, Ruo Ando, Takeshi Takahashi, Satoshi Uda, and Tomoya Inoue. A measurement study of open resolvers and dns server version. In Internet Conference (IEICE), 2013.

Zakir Durumeric, Eric Wustrow, and J. Alex Halderman. Zmap: Fast internet-wide scanning and its security applications. In Proceedings of the 22th USENIX Security Symposium (Security), pages 605–620, Washington, DC, August 2013.

Luuk Hendriks, Ricardo de Oliveira Schmidt, Roland van Rijswijk-Deij, and Aiko Pras. On the potential of ipv6 open resolvers for ddos attacks. In Proceedings of the 18th International Conference on Passive and Active Measurement PAM, pages 17–29, 2017.

Lanlan Pan, Xuebiao Yuchi, and Yong Chen. Mitigating ddos attacks towards top level domain name service. In Proceedings of the 18th Asia-Pacific Network Operations and Management Symposium APNOMS, pages 1–4, 2016.

Zhen Qin, Chunjiang Xiao, Qiyao Wang, Yuehui Jin, and Aleksandar Kuzmanovic. A cdn-based domain name system. Computer Communications, 45:11–20, 2014.

Benjamin Greschbach, Tobias Pulls, Laura M. Roberts, Phillip Winter, and Nick Feamster. The effect of DNS on tor’s anonymity. In Proceedings of the 2017 Annual Network and Distributed System Security Symposium (NDSS), San Diego, CA, February–March 2017.

Kyle Schomp, Michael Rabinovich, and Mark Allman. Towards a model of DNS client behavior. In Proceedings of the 25th USENIX Security Symposium (Security), pages 263–275, Austin, TX, August 2016.