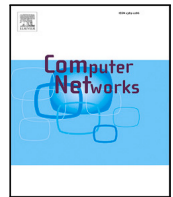




Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet



Domain name system security and privacy: A contemporary survey

Aminollah Khormali, Jeman Park, Hisham Alasmay, Afsah Anwar, Muhammad Saad, David Mohaisen *

University of Central Florida, United States of America

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 Internet Protocol (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 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 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 [1–5], 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 [3], 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. Towards 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.

* Corresponding author.

E-mail addresses: aminkhormali@knights.ucf.edu (A. Khormali), parkjeman@knights.ucf.edu (J. Park), hisham@knights.ucf.edu (H. Alasmay), afsanahwar@knights.ucf.edu (A. Anwar), mohaisen@cs.ucf.edu (D. Mohaisen).

<https://doi.org/10.1016/j.comnet.2020.107699>

Received 26 June 2020; Received in revised form 29 October 2020; Accepted 18 November 2020

Available online 2 December 2020

1389-1286/© 2020 Published by Elsevier B.V.

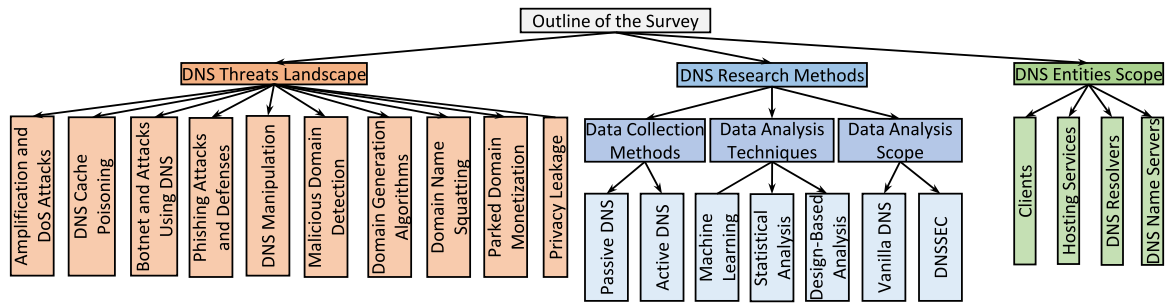


Fig. 1. The general outline of the survey. We have surveyed a wide variety of research works from various points of view, including DNS threat landscape, DNS research methods, and DNS research scopes.

1.1. Outline

The general outline of the survey, highlighting the road map towards the key terms and directions, is shown in Fig. 1. In the following we briefly review some of those broad directions, and will describe each of them in the rest of the paper.

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 grown, however, adversaries have found weaknesses that can be abused for malicious purposes. There is a large body of research in this area, which has led to the identification of major threats that substantially compromise the DNS security. In DNS Threat Landscape (Section 3) 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* (Section 3.1), *DNS Cache Poisoning* (Section 3.2), *Botnet and Attacks Using DNS* (Section 3.3), *Phishing Attacks and Defenses* (Section 3.4), *DNS Manipulation* (Section 3.5), *Malicious Domain Detection* (Section 3.6), *Domain Generation Algorithms* (Section 3.7), *Domain Name Squatting* (Section 3.8), *Parked Domain Monetization* and associated security issues (Section 3.9), and *Privacy Leakage* (Section 3.10).

DNS Research Methods. Another vertical of our survey is the DNS Research Methods (Section 4) 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 (PDNS)* (Section 4.1.1) and *Active DNS (ADNS)*. Next, we categorize the research works based on the common DNS Data Analysis Techniques that have been used in the literature, such as *Machine Learning Algorithms* (Section 4.2.1) and *Association Analysis* (Section 4.2.2). Finally, we categorize the research works based on their DNS Data Analysis Scope and from security point of view into *Vanilla DNS* (Section 4.3.1) and *DNS Security Extensions (DNSSEC)* (Section 4.3.2).

DNS Entities Scope. In DNS Entities Scope (Section 5) we categorize the literature based on the DNS entities they target, including *DNS Name Servers* (Section 5.1), *Open DNS Resolvers* (Section 5.2), *Hosting Services* (Section 5.3), and *Clients* (Section 5.4). Such categorization would reveal the most susceptible entity and its impact on the performance and security of whole DNS ecosystem.

1.2. Contribution

We discuss advantages and disadvantages of the proposed countermeasures at the end of each section and highlighted the existing open problems. We list the common abbreviations used throughout this paper in Table 1. Finally, the key characteristics of our survey in this paper can be summarized as:

- We conduct a comprehensive survey mainly focusing on recent major advances in the area of DNS security and privacy. The focuses of the survey are the threat landscape, research methods, and research scopes (based on a classification of the DNS system model).
- The scope of the survey is defined in such a way that not only it focuses on the up-to-date challenges, by gathering works mostly published in the past decade, but also by focusing on the most critical and ongoing challenges corresponding to the current trend.
- We highlight the existing open challenges in the security and privacy of DNS infrastructure, which requires further investigations by security specialists to be addressed. We use that as a summary in every surveyed area.

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 [6] 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.

Table 1
List of abbreviations in alphabetical order.

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

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 Fig. 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 ② through ⑦, to answer the query. The root server is the first server that receives queries from recursive resolvers, in step ②. The root servers are globally distributed and maintain the IP addresses and location of authoritative TLD name servers. In step ③, the root name server replies to the query with the appropriate list of authoritative TLD servers for the .com TLD. In step ④, 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 ⑤. At the next step, namely step ⑥, 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 ⑦ 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

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. By outlining the key threat landscape elements, we hope to bring the attention of the community to the lacking defense mechanisms in place, which would require further exploration.

In this section, we would delve into the common threats in the DNS ecosystem and state-of-the-art countermeasures presented in the literature. For example, attackers abuse the functionality of open DNS

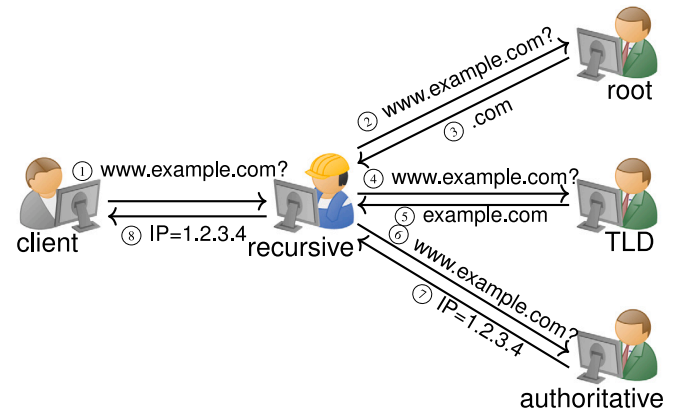


Fig. 2. Illustration of DNS resolution over recursive, root, TLD, and authoritative name server. Once a user attempts to access www.example.com (step ①), in the case that there is no information about the given domain in the local cache and host table, recursive resolver starts asking root, TLD, and authoritative name servers and over steps ② through ⑦ to answer the query.

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.

• **Amplification and DoS Attacks:** Despite the large body of research works on detection and mitigation of *DNS amplification attacks* [7–15], such attacks still compromise today's Internet, according to *Nexusguard's* 2018 threat report of Q1 [16]. • **DNS Cache Poisoning:** 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* [17–23] using different methods, such as DNSSEC [17,24], Wildcard SECure (WSEC) DNS [20], Expected Consistency Optimization for DNS (ECO-DNS) [19], and Anax [18], among others, which we outline this study. • **Botnet and Attacks Using DNS:** 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 [25], amplification attacks, and credentials stealing, among many others [26]. 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 [13, 27–30], 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 attacks IoT devices, such as routers and video recorders [31,32], and uses them for amplification attacks, necessitating security solutions optimized for IoT and its interplay with DNS [33,34].

- **Phishing Attacks/DNS Manipulation:** DNS manipulation may expose users to multiple threats, such as phishing and malicious domains. Accordingly, a large number of studies considered DNS manipulation [35–39], and its interplay with the detection and mitigation of *phishing attacks* [40–45]. 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 [46], highlighting the need for further investigation and defenses.
- **Malicious Domain Detection:** A large body of research has been devoted to the detection of *malicious domain names* [27,40,47–59], which can be classified into two general categories: classification based [27,40,47–49] and inference based [52,55,56,59] methods. 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:** 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 [13,29,60–63].
- **Domain Name Squatting:** 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 [5,64–70], bitsquatting [71–73], combosquatting [42], and soundsquatting [74]. Domain name squatting becomes popular as an attack vector to start many other types of attacks [66,67].
- **Parked Domain Monetization:** 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 [67,75–77].
- **Privacy Leakage:** 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, such as EncDNS [78], Privacy-Preserving DNS (PPDNS) [79], DNSSEC Look-aside Validation (DLV) [80], etc.

In addition, we discuss challenges and highlight open problems that require further investigations by security researchers associated with each of these major threats, each of which can cause disastrous impact on the operation of whole infrastructure. In Tables 9 and 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 Fig. 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

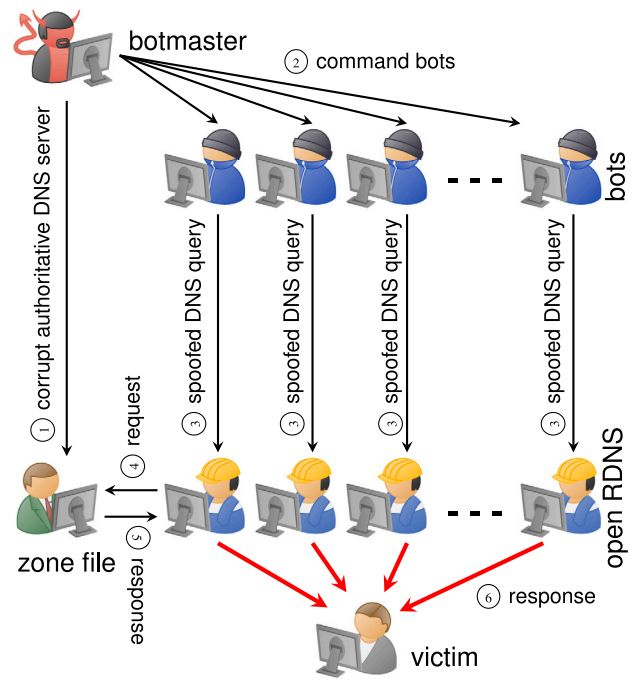


Fig. 3. High-level architecture of a typical DNS amplification attack. Small number of DNS queries are being transformed into a significantly large payload coordinated at the target network.

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 [7–15,81,82]. For instance, Ballani et al. [8] 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. [10] 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. [82] 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 [81] 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. [11] examined a large number of domains ($\approx 129M$) and authoritative servers ($\approx 1.1M$) to investigate the inherent DNS amplification risks associated with DNS authoritative name servers. Their analysis showed that only a small number of authoritative servers ($\approx 3.8\%$) are responsible for the highest amplification factors. In addition, their analysis revealed that adoption of DNS response rate limit is limited to less than $\approx 3\%$ of authoritative servers. Finally, they have suggested tunneling into a remote resolver as a

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		
			RL	ID	$O^{(1,2,3)}$
[7]	sFlow/SDN	Near real-time detection	✗		
[8]	Stale cache	Simple, lightweight			$\chi^{(1)}$
[9]	One-to-one mapping	Complete mitigation	✗		
[10]	DNS authentication	Cloud based service	✗		
[11]	Tunnel/remote resolver	Legacy compatible	✗		$\chi^{(2)}$
[13]	ML/domain features	Small feature space			$\chi^{(3)}$
[14]	DRS-ADAM	Easy deployment		✗	
[15]	RADAR	Real-time detection		✗	

straightforward and simple countermeasure to mitigate on-going attack at the organization level. Verma et al. [14] 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. [7] have incorporated sFlow with security-centric SDN features to analyze DNS query identifiers (IDs) for detecting and mitigating DNS amplification attack 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. [9] 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. Erhan et al. [83] have presented a new DDoS detection approach employing the Matching Pursuit algorithm for detection of resource depletion type DDoS attacks. Their proposed algorithm is able to detect low-density DDoS attacks.

Zheng et al. [15] 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. [13] 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.

Recently, Lu et al. [84] have presented range-based amplification attacks, which allows attackers to overwhelm both outgoing bandwidth of the origin servers deployed behind CDNs and bandwidth of CDN surrogate nodes. Furthermore, Chang et al. [85] have presented a query-crafting attack where the attacker is able to control the DNS query payload to increase the threat impact per query.

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 proposed methods, which would

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		
			LDR	AR	WSR
[17]	DNSSEC	A-posteriori detection	✗		
[18]	Anax	Easy deployment		✗	
[20]	WSEC DNS	Easy deployment			✗
[24]	DNSSEC	Cryptographic/verifiable	✗		

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 [7,9–11]. 2. requiring a range of changes to the DNS resolvers and semantics [8], 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 [12], or requiring large number of flow rules [15], which calls for aggregate and light-weight feature engineering methods. 4. low detection accuracy, in some cases [13], which calls for improving accuracy through multi-modality 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 Fig. 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 [17–23].

Herzberg and Shulman [24] studied the security of the patched DNS and found that not only source ports may be circumvented by various Network Address Translation (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. [35] have analyzed a large number of measurement sessions from distinct IP addresses, collected by Netalyzr [86], 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 Internet Service Provider (ISP) driven manipulation of DNS.

Shulman and Waidner [17] studied the security of DNS infrastructure and highlighted that although adoption of challenge–response defenses [87,88] 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. [18] 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 analyzes resource records and extracts a set of statistical features, such as domain name diversity, 2LD diversity,¹ 3LD diversity

¹ xLD refers to the x-th level domain. In <http://www.example.com>, example.com is a 2nd level domain, and www.example.com is a 3rd level domain.

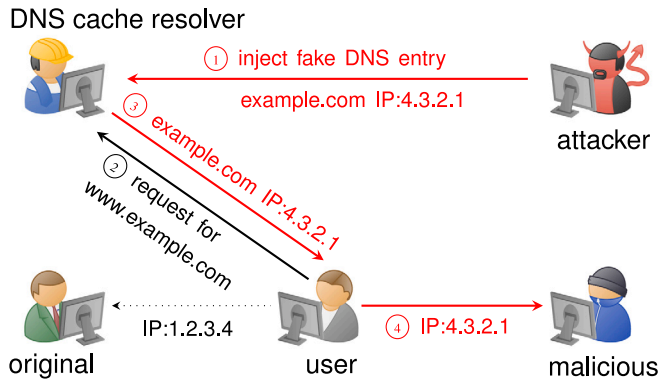


Fig. 4. DNS cache poisoning attack. Attacker injects fake DNS entries into the DNS cache resolvers in an attempt to direct users into fake destinations, mainly for malicious purposes like phishing.

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. [20] proposed 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. [21] 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. [22] 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 increase, 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 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 [23] 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.

Alharbi et al. [89] have proposed a new type of DNS cache poisoning attacks that targets the client-side DNS cache. The proposed attack allows an off-path attacker to collaborate with a piece of illegitimate malware to poison client machine. Furthermore, in a very recent research work, Zheng et al. [90] have presented a novel cache poisoning attack targeting DNS forwarding through injecting rogue records of arbitrary victim domain names. They have tested this attack on several home router models and found that their proposed attack is able to circumvent large vendors such as D-Link, Linksys, etc.

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 Table 3. The limitations of proposed solutions can be summarized as: 1. low

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	
			AR	LDR
[13]	ML/domain length	Small feature space	✗	
[28]	DNSSEC	Strict delegation		✗
[28]	ML/statistical features	Simple	✗	
[30]	Probability/DNS traffic	Simple	✗	

deployment rate of DNSSEC [17,24], 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 [18, 23], calling for better accuracy using multiple types of features. 3. causing overheads on DNS traffic, memory usage and increasing response latency [20], calling for lightweight and more efficient approaches. 4. and wasting DNS cache storage resources by prefetching unpopular resource records [19], which calls for adaptive resolution.

3.3. Botnet and attacks using DNS

The word botnet is a combination of the words robot and network, and refers into a number of Internet-connected devices, each of which is running one or more bots. Botnets are composed of various infected hosts, C&C channels, and a botmaster. The general infrastructure of a botnet is shown in Fig. 5. Botnets can be used to perform DDoS attack [25], password theft [26], and allow the attacker (botmaster) to access devices and their connection through C&C channels. Researchers have investigated the threat of botnets various works, including the intersection between botnets and DNS [13,27–30].

Jiang et al. [28] have studied whether deleting malicious domain names from upper level DNS servers can prevent botnet C&C and malware propagation. Their analysis showed that ghost domain names stay resolvable even long after the delegated data has been removed from the domain registry and the record's TTL is expired. Finally, they suggest adoption of DNSSEC to overcome the threat of ghost domain attack.

Jin et al. [27] 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. [91] have performed longitudinal study to identify novel botmaster strategies through analyzing multiple active botnet families.

Truong and Cheng [13] 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 [92] and Zeus [93]. 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. [30] 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.

Vinayakumar et al. [94] have presented a two-stage deep learning based framework for botnet detection at the application layer of the DNS services. While in the first stage the siamese network is utilized to measure the similarity of DNS queries across Ethernet connections, in the second stage a DGA based on deep learning structure is recommended to identify normal and malicious domain names. Moreover,

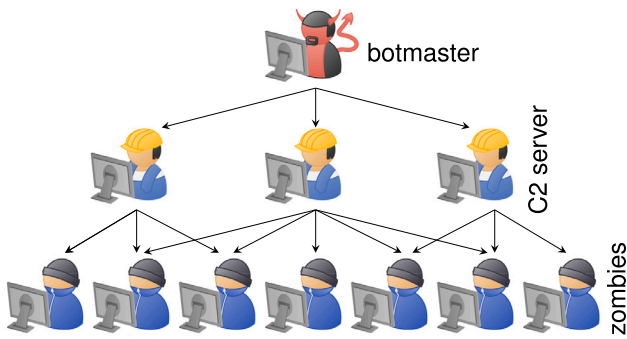


Fig. 5. An overview of a botnet infrastructure. Attacker (botmaster) remotely abuses infected devices (zombies) and its connections through C2 channels for malicious activities.

Wang et al. [95] have proposed BotMark, an automated botnet detection with hybrid analysis of flow-based and graph-based traffic behaviors. It requires no prior knowledge of botnets and is independent of C&C protocols and structures.

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 [13,28], high false alarm rate [30], and low deployment rate of DNSSEC [28]. 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 [31,32] highlighting the need for security solutions optimized for the IoT devices [33,34]

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 Municipal Collection Services (MCSI) Inc. 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 [96] only at one particular country, Singapore. Therefore, researchers have conducted a large number of works to detect and mitigate phishing attacks [40–44].

For example, Bilge et al. [40] 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. [43] 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. [44] presented a phishing detection method based on the visual-similarity that works based on analyzing three main

Table 5

Summary of sample research works that have studied phishing attacks. Here G is Generic, L is Lightweight, and AR donates Accuracy Rate. The common weakness of these studies is low detection accuracy.

Work	Method	Advantages		Disadvantages
		G	L	
[40]	EXPOSURE	✓		AR
[43]	Phishwish		✓	AR
[44]	Visual similarity		✓	AR
[45]	Visual similarity	✓		AR

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. [45] 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.

In a more recent work, Oest et al. [97] have proposed Golden Hour, a framework for not only passive collection of victim traffic to phishing pages. Golden Hour was able to record 4.8 million traffic records who visited phishing pages, excluding crawler traffic.

Discussion and Open Directions. 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 [43,45], requiring large number of train samples [40], dependency of the performance to train inputs [40,44], and not being adaptive to the changes in the scenarios [45] 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 [46], 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 [36–39].

For instance, the vulnerability of the user-side DNS infrastructure to record injection threats have been measured by Schomp et al. [36]. 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 [98] and the deployment of familiar defensive methods, e.g., 0x20 encoding [99].

Kuhrer et al. [37] have investigated the negative aspects of open DNS resolvers that can be abused 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. [100] have proposed an approach for the detection of unauthorized DNS root manipulation using two different techniques. The first

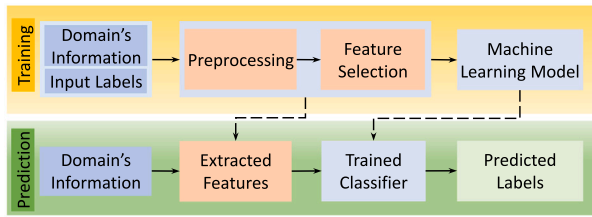


Fig. 6. General flow of the malicious domains detection based on machine learning algorithms. It is composed of two steps; training ML model over representative features of labeled domain names and then using the trained model to predict unknown domain names.

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. [38] 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. [39] 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 and Open Directions. DNS infrastructure suffers from multiple issues, e.g., inefficient look-ups, unreliability of IP-level fragmentation, and ISP-driven DNS manipulation [35]. Moreover, record injection vulnerabilities, e.g., Kaminsky [98] are widespread [36]. Attackers can utilize standard-conforming resolvers to circumvent mitigation techniques e.g., source port randomization as well as IP address randomization [24,39] to manipulate DNS. In addition, ISPs are clearly willing to involve in DNS manipulation for reasons, such as error traffic monetization [76]. 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 [100]. All of those observations highlight the need for employing various mechanisms that take them into account, individually, for better defenses to improve the security of DNS.

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 [27,40,48–54,56–59,101,102].

This work is classified into two general categories, classification based [27,40,47–49] and inference based [52,56,59]. 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 Fig. 6.

Table 6

Summary of the research works that have investigated malicious domain detection. Here G is Generic, L is Lightweight, P is Proactive, AR is Accuracy Rate, E is Evadable, DID is Dependency on Input Data, and LV denotes Local Visibility.

Work	Method	Adv.			Disadv.			
		G	L	P	AR	E	DID	LV
[27]	ML/statistical		✓		×			
[40]	EXPOSURE	✓				×	×	×
[47]	Notos			✓	×		×	×
[48]	Kopis		✓				×	
[49]	Ext. EXPOSURE		✓			×	×	×
[50]	Prophiler		✓		×			
[51]	Proactive blacklisting			✓	×			
[52]	Temporal correlation	✓				×		
[53]	PREDATOR			✓	×	×		
[56]	Syntactic/temporal	✓				×		
[59]	Woodpecker			✓		×		

Classification based approach. Antonakakis et al. [47] 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, Border Gateway Protocol (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. [48] presented Kopis, which detects malware-related domains by monitoring DNS traffic at the upper DNS hierarchy. Kopis 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. Kopis 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. [40] 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 [49]. They showed that EXPOSURE is able to make meaningful connections between various malicious domains. For example, there exist IP addresses that are shared among thousands of unique domain names indicative of botnet-related domain names.

Jin et al. [27] 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.

Sun et al. [101] proposed HinDom that is able to detect malicious domains with very limited number of labeled data thanks to the metapath-based transductive classification method. It constructs a heterogeneous information network (HIN) of clients, domains, IP addresses and their diverse relationships, which in turn provides a global viewpoint. The same authors have proposed DeepDom [102], which is

a similar approach as HinDom, however DeepDom uses graph convolutional network to detect malicious domains using joint consideration of both domain's local features and their global associations.

Inference-based approach. Strong associations of domains with known malicious domains is a vital point in effective malicious domain detection. Therefore, Khalil et al. [55] have designed an association-based scheme for the detection of malicious 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. [52] 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. [56] have proposed a comprehensive framework that uses evolutionary learning for detection of diverse clusters of DNS failures, e.g., highly random, mutated string, and sub-string 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. [54] 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. [53] 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, CPM that detects malicious domains before they appear in DNS blacklists. Moreover, Felegyhazi et al. [51] 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. [57] 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. [50] 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 and Open Directions. 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 [12]. 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

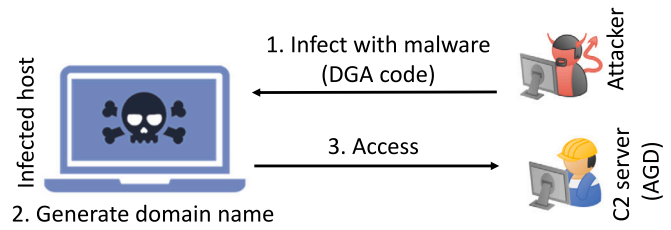


Fig. 7. The General overview of the communication of the malware-infected hosts with C&C servers. DGAs are being used as main method of communication in botnets.

```
1 def generate_domain(y, m, d):
2     domain =
3     for i in range(16):
4         y = ((y^8*y)>>11)^((y&0xFFFFFFFF)<<17)
5         m = ((m^4*m)>>25)^16*(m&0xFFFFFFFF8)
6         d = ((d^(d<<13))>>19)^((d&0xFFFFFFFFE)<<12)
7         dm+=chr(((y^m^d)%25)+97)
8     return domain
```

Fig. 8. A sample DGA code that is utilized to intermittently generate large number of algorithmic domain names, mainly for malicious purposes [63].

accuracy [27,47,50,51,53], local visibility [40,47,49], expensive computation [57], dependency on training dataset [40,47–49], requiring correlation between anchor malicious domain and unknown malicious domains [52,56], and can be evaded by avoiding certain features or behaviors [40,49,53]. In turn, the current state of the literature in malicious domain name detection, and the fact that they are still prevalent as a main threat vector, calls for new directions detection that aim to address those issues and shortcomings of the prior work.

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, Fig. 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, Fig. 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 [13,29,60–63,103–105].

Yadav et al. [60] studied algorithmically generated domain names for domain fluxing. Statistical measures, such as Kullback–Leibler divergence [106], 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.

Plohmman et al. [29] 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 of C&C accesses as well as accurate identification of malware families and related campaigns of future DGA domain names.

Antonakakis et al. [61] have investigated unsuccessful DNS resolutions to detect groups of potential DGA domains. They built a DNS-based DGA domain detection system, called Pleiades, which works

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
	T	B	C	SS	PAF	S	HS	AM		
[5]	✓	✓		✓	✓	✓	✓	✓	Squatting comparison	–
[42]			✓					✓	Combosquatting	No detection
[68]	✓				✓				Typosquatting detector	Large user traffic
[64]	✓				✓				Typo-generation models	Limited data
[66]	✓				✓	✓			Longitudinal study	Limited data
[67]	✓				✓			✓	YATT mitigation tool	Updating blacklist
[71]		✓			✓	✓		✓	Bitsquatting/defense	impractical
[72]		✓			✓	✓		✓	Bitsquatting/defense	Costly/impractical
[73]	✓	✓				✓			Name server squatting	Low deployment
[74]				✓	✓		✓	✓	Soundsquatting	Accuracy rate

based on the characteristics of partially registered algorithmically generated domain names.

Spaulding et al. [63] have presented a malicious domain detection system that proactively detects algorithmically generated domain names. Their work highlighted the difference between the number of NXDomain responses of benign and algorithmically generated domain names prior to registration. These patterns are then used to build a classification system using sliding time windows. Barabosch et al. [62] have presented an automatic method to extract DGAs from malware binaries using dynamic analysis in combination with data flow analysis. In addition, a taxonomy of DGA families is defined using time dependency and causality features. Truong et al. [13] have analyzed DNS traffic to design a detection system for recognition of algorithmically generated domain names from legitimate domain names. The length of domain names and their expected values are then used to build a classifier that identifies pseudo-random domain names generated by botnets, such as Conficker [92] and Zeus [93].

Almashhdani et al. [103] presented MaldomDetector, a system for malicious domain names built using machine learning algorithms. MaldomDetector utilizes domain name's characteristics to detect DGA-based communications and circumventing the attack prior to making any successful connection with the C&C server. Furthermore, Xu et al. [104] have developed a method to detect DGA-based domain names using the combination of n-grams and convolutional neural network. In a similar work, Yu et al. [105] have investigated the use of deep neural networks for DGA-based domain name detection. Their method is based on the characteristics of the domain names, trained large number of heuristically labeled real traffic.

Discussion and Open Directions. Some major challenges of detecting DGA-based domains include constructing a taxonomy of DGA families [29,62], collecting domain names generated by different malware families [29], reconstruction of DGAs by reverse engineering malware [61,62], learning newly found DGAs (being generic) [60,61], and low detection accuracy [13]. All of those issues would require further attention through various research efforts, including an updated explorations of the issue through new measurements to track and understand evolution in this aspect of the attack surface, and new mechanisms that would address the various shortcomings highlighted above and associated with the current state-of-the-art techniques.

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. Cybersquatters register variants of popular trademark names through different squatting strategies, including typosquatting, bitsquatting, 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 [42, 66–70,73], 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 [64,65]. For instance, Wang et al. [64] 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. [65] 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. [5] have explored the landscape of the typosquatted domain names and summarized common typo-generation models and monetization strategies, which are shown in Fig. 9.

Agten et al. [66] 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 typosquatted domains. Furthermore, they observed a change in the trend and monetization behaviors of typosquatters over time and through hosting different types of websites, e.g., parked/ads/for sale domains and scams. Moreover, Szurdi et al. [67] 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. [65] 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 [66,67] are in contrast to the results presented by

Khan et al. [68] 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 s per typosquatting event over the alternative of receiving a browser error page.

Spaulding et al. [69,70] have conducted a user study to understand the effectiveness of the typosquatting strategies in deceiving users. The behavior of users, who have been exposed to several typosquatting URLs, have been examined in an attempt to figure out whether their

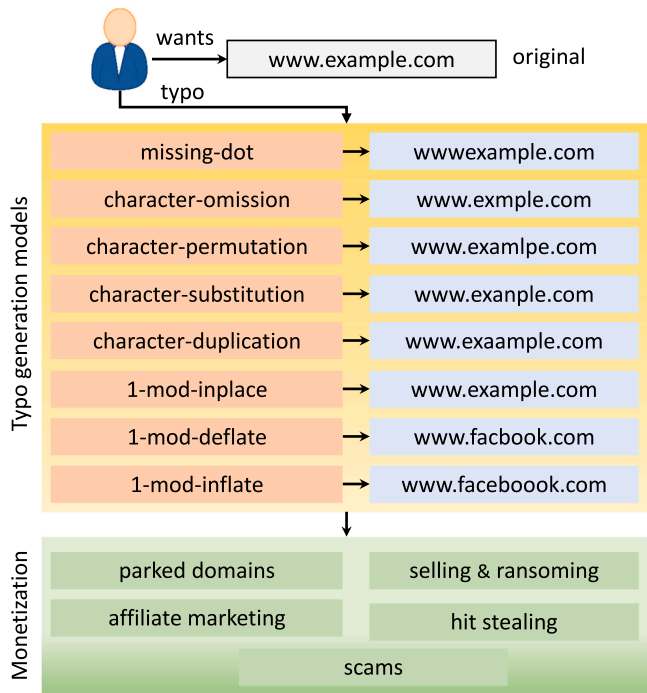


Fig. 9. Common typo-generation models and monetization strategies. Notice that permutations and substitutions of characters are among the most effective techniques in deceiving users.

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 [71,72] 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 [71] through investigating all HTTP requests to 30 registered bitsquatted 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. [72] 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 pre-registering all possible bitsquatting domains.

Vissers et al. [73] 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. [42] 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 large fraction of them ($\approx 60\%$) 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 *whether*, *weather* and *idle*, *idol*, *idyll* to generate squatted domains. Nikiforakis et al. [74] 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 [111], ORCA [112].

3.8.5. App squatting

The app squatting is a threat associated with the mobile app ecosystem, where the attacker releases a malicious application whose identifiers are similar to that of well-known apps. Hu et al. [113] have conducted a comprehensive analysis of app squatting where they were able to generate approximately 20 malicious apps for each of the top 500 apps in google store.

Discussion and Open Directions. 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 [42,67]. The limitations of the existing research proposals can be summarized as following: (1) lack of practical yet automatic domain squatting detection system [42], (2) requiring a huge amount of user traffic or content for building models [68], and (3) lack of practical yet effective mitigation techniques [71–73]. For instance, the research proposals which suggest deployment of DNSSEC [72,73] should consider the fact that deployment rate of DNSSEC although increasing but still far from ideal point, thus impractical in large-scale and devising incentives to facilitate this deployment as a solution. The current state-of-the-art advances, and associated limitations, call for further work in understanding why squatting techniques work in practice, whether it is due to users perception (or lack of it), system issues, and techniques to address them. Moreover, as squatting is technology-dependent, our literature review highlight the need for exploring various avenues of this attack surface in new and emerging technologies, such as IoT and virtual reality (VR), how such attack avenue is facilitated by the different input mechanisms in these technologies, and methods that could be used for defending against the attack using the constraints of those environments/technologies.

Table 8
Summary of the research works that have studied DNS privacy landscape.

Work	Method	Strengths	Weaknesses
[78]	EncDNS	Lightweight/Low-latency	Unresolvable internal domains
[79]	PPDNS	Improved privacy	Complex/DNS modification
[80]	DLV-Aware/PP-DLV	Improved privacy	MITM attacks/record modification
[107]	Range query	Simple/flexible	Bandwidth/response latency
[108]	PIR	Improved bandwidth	Protocol modification
[109]	T-DNS	Improved privacy/security	Response latency
[110]	No DNS prefetching	Inference attacks	Inaccurate TTL/Unstable profiles

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; 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 [67,75–77].

Alrwais et al. [75] 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. [67] 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. [76] have investigated the error traffic monetization practices of ISPs through analyzing 66,000 *Natalyzr* session traces. Their analysis showed that ISPs were clearly involving on error traffic monetization by rerouting traffic to ad servers.

Vissers et al. [77] 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 and Open Directions. 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. [75, 77]. 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 [75]. Moreover, domain monetization is prevalent among both popular domains and less known domains [67]. In addition, ISPs actively involve traffic monetization activities [76]. One open research direction is to design a detection and mitigation system that can effectively defend users against the threats of parked domains.

3.10. Privacy leakage

A range of works have been conducted to address privacy-related issues of DNS [78–80,109,110,119–122,127]. For example, Herrmann et al. [78] 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 encapsulation of encrypted messages in standards-compliant DNS messages. EncDNS is compatible with existing popular DNS resolvers and offers low-latency DNS resolution. Furthermore, Shulman [128] 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. [110] 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. [109] 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. [129] 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 proposed mechanism reduces the DTLS round trips and the handshake size as well.

Zhao et al. [107] 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 García-Alfaro [119] 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 García-Alfaro [119] have evaluated two DNS privacy-preserving proposals, the *Range Query* [107] and the *Privacy Information Retrieval (PIR)* schemes [108]. 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 [79] explored DNS privacy leaks during domain resolution and presented 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, Ferrerath et al. [130] argue that although PPDNS provides high level of privacy, it would not be adopted in the near future due to its

Table 9

Topical classification threats addressed in the literature, with sample work. ADA=Amplification and DoS Attacks, DCP=DNS Cache Poisoning, BAD=Botnet and Attacks Using DNS, PAD=Phishing Attacks and Defenses, DM=DNS Manipulation, MDD=Malicious Domain Detection, DGA=Domain Generation Algorithms, DS= Domain Name Squatting, PDM=Parked Domain Monetization, PL=Privacy Leakage.

Work	ADA	DCP	BAD	PAD	DM	MDD	DGA	DS	PDM	PL
[1]										✓
[48]	✓		✓	✓		✓				
[114]				✓						
[13]	✓		✓				✓			
[12]	✓									
[115]	✓									
[49]						✓				
[42]				✓				✓	✓	
[68]								✓		
[23]		✓								
[52]						✓				
[55]						✓				
[18]		✓								
[47]						✓				
[54]						✓	✓			
[56]						✓				
[57]						✓				
[116]										
[117]		✓					✓			
[77]								✓	✓	
[60]	✓		✓				✓			
[36]					✓					
[41]				✓						
[30]	✓		✓							
[19]		✓								
[118]		✓								
[24]					✓					
[10]	✓									
[8]	✓									
[78]										✓
[37]					✓	✓				
[38]					✓					
[20]		✓								
[14]	✓				✓					
[7]	✓									
[11]	✓									
[17]		✓								
[21]		✓								
[22]		✓								
[27]			✓			✓				
[28]	✓		✓			✓			✓	
[29]			✓				✓			
[35]					✓					
[40]	✓		✓	✓		✓				
[50]						✓				
[51]						✓				
[53]						✓				

computational complexity and requirement for completely different DNS infrastructure.

Table 10

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

Work	ADA	DCP	BAD	PAD	DM	MDD	DGA	DS	PDM	PL
[58]						✓				
[59]							✓	✓		
[67]								✓	✓	
[73]								✓		
[75]									✓	
[76]					✓				✓	
[79]										✓
[80]										✓
[100]					✓					
[109]					✓					
[110]										✓
[119]										✓
[120]										✓
[121]										✓
[122]										✓
[123]	✓									
[124]						✓				
[125]						✓				
[126]		✓								

DNS blocking of unintended queries is among the suggested solutions to improve the DNS privacy. For example, Appelbaum and Muffett [131] have studied Tor's privacy and suggested to block .onion names at stub, recursive, and authoritative resolvers to improve its privacy. Mohaisen and Ren [121] 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. [122] 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 and Open Directions. 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 [107]. Privacy issues have been overlooked by DNS security efforts (such as DNSSEC) and are thus likely to propagate into future versions of DNS [79], requiring further attention.

With the deployment of various efforts to make DNS more secure and private in the recent years, it is of paramount importance to understand the dynamics and interactions between different components that make up the Internet and how they affect the privacy guarantees of those approaches.

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 Fig. 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

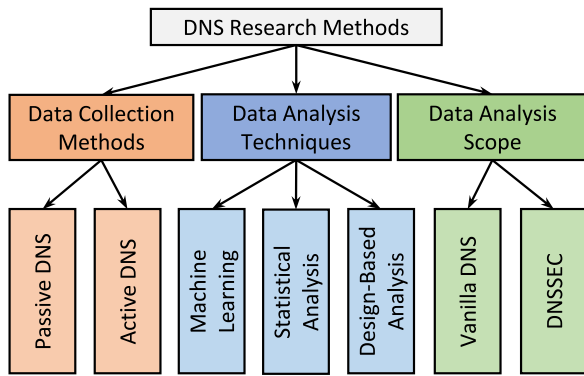


Fig. 10. DNS Research Methods. We have categorized DNS research method into data collection methods, data analysis techniques, and analysis scope.

for future investigations. In particular, first, we describe two main DNS Data Collection Methods utilized in the literature and the associated works, including *Passive DNS (PDNS)* [12,13,23,42,48,49,52,68,114,115,132–134] and *Active DNS (ADNS)* [42,55,68,134–136]. Second, we categorize the research works from the DNS Data Analysis Techniques point of view into *Machine Learning Algorithms* [18,23,47,54–57,77,116,117,132,137] and *Association Analysis* [55,60,114,138,139]. Finally, we group the research works based on the DNS Data Analysis Scope and from security point of view into *vanilla DNS* [19,30,36,41,99,140,141] and *DNS Security Extensions (DNSSEC)* [10,24,118,136,142–146].

4.1. Data collection methods

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

4.1.1. Passive DNS

Florian Weimer [147] 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 [148]. High-level architecture of passive DNS measurement systems is shown in Fig. 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 [12,13,23,42,48,49,52,68,114,115,132–134].

For example, Callahan et al. [133] have analyzed 200 million DNS queries collected through passive monitoring of DNS traffic of 90 home residential neighborhood network in the U.S. [133] in order to investigate the evolution and the behavior of the DNS servers and clients in modern DNS systems. Moreover, Gao et al. [52] have analyzed 26 billion DNS query–response pairs collected by Security Information Exchange (SIE), now part of DNS database (DNSDB) [149], from more than 600 recursive DNS resolvers distributed over North America and Europe, to empirically reexamine 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 [23] 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. [132] 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.

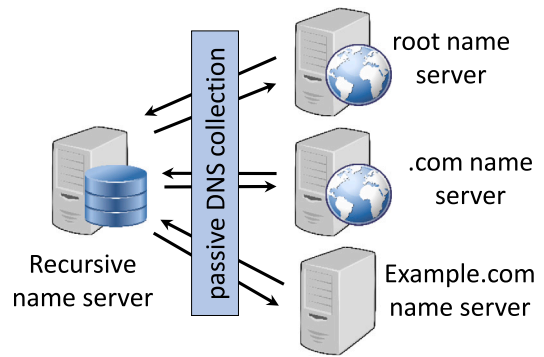


Fig. 11. High-level architecture of passive DNS collection system. In passive DNS collection, the DNS requests and responses are gathered from DNS providers across the internet as they happen.

Liu et al. [59] have queried passive DNS data from Farsight Security passive DNS database [149] and the biggest public passive DNS system in China [59] to detect malicious domains. Moreover, Tajalizadehkhoob et al. [114] have analyzed 1259 shared hosting providers extracted from DNSDB [149] to explore the distribution of web security features and software patching practices in shared hosting providers.

Furthermore, Khan et al. [68] 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. [42] 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. [49] have analyzed the passive DNS data obtained by SIE [150] to detect and block malicious domain names in real time. In addition, Li et al. [115] have investigated the topological relation of hosts and malicious websites using the same PDNS API, SIE [150]. Moreover, Perdisci et al. [12] have utilized the SIE PDNS traffic to detect and track malicious domain-flux networks. Similarly, Truong et al. [13] 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. [13,23,52,68], the collection of such data is costly, challenging, and needs restrictive legal agreements [134].

4.1.2. Active DNS

The concept of active DNS measurement have 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 [134]. 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 [42,55,68,100,134–136,151]. 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 Fig. 12.

Kountouras et al. [134] 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 [152–154], the Alexa list [155], the Common Crawl dataset [156],

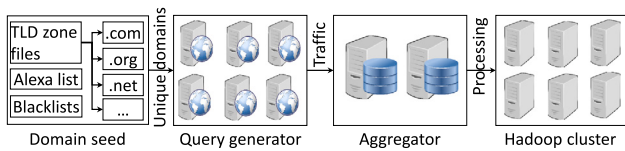


Fig. 12. High-level architecture of a typical active DNS collection system. Active DNS collection is composed of four main steps including gathering domain seeds, generating queries, collecting responses, and storing/processing the data in Hadoop cluster.

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. [135] 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. [68] have actively crawled 13.5k suspected domains to study the impact of typosquatting events on both typical users and legitimate websites. Kintis et al. [42] have analyzed 455 billion DNS RRs collected daily through the active DNS project [134,157] 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. [55] have conducted an in-depth analysis of active DNS dataset provided by Thales [134,157] to design an association-based scheme for detection of malicious domains with high accuracy and coverage.

Chung et al. [136] have collected all DNSSEC-enabled domain names, under the .com, .org, and .net TLDs through OpenINTEL active DNS measurement system [135], to study the adoption and management of DNSSEC in both authoritative name servers and DNS resolvers. Moreover, Toorn et al. [158] have studied the detection of Snowshoe spams through active DNS measurement and using machine learning techniques.

Discussion and Open Directions. 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. In addition, issued DNS queries based on the limited set of hosts yield a localized DNS data collection. In other words, the collected DNS data is limited to the locations of those hosts, thus all IPs associated with a specific given name might not be identified. Moreover, DNS queries are made by data collectors not by individual users, which makes active DNS data less suitable as an approach to detect malicious domains based on user level features of DNS data, such as user query patterns. The summary of the collected dataset in the literature is in Table 11 along with the query types description in Table 12.

As most of the findings and conclusions regarding the security and use landscape of DNS, using either method of data collection, would significantly depend on the time in which the data is collected, there is a constant need for new datasets that consider both (and potentially other hybrid) approaches. While also important, it seems that the prior work using those methods have overlooked query loads and application contexts of the data collection, making it important to investigate the applicability of those methods in a range of contexts (e.g., enterprise, home, and backbone data collection).

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 [18,59,132]. This section introduces some of the prevalent data analysis techniques that have been used in the cybersecurity community, such as machine learning algorithms [18,23,47,54–57,77,116,117,132,137] and association analysis [52,55,60,138,139].

4.2.1. Machine learning algorithm

Antonakakis et al. [18] 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 [23] have trained decision tree and random forest 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. [132] 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. [55] 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. (see Tables 13–15).

Liu et al. [59] 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. [77] 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. [159] have used TSVM algorithm instead of classical SVM to improve the detection accuracy for phishing webpages.

One of the advantages of the machine learning algorithms is their ability to recognize complicated patterns, where defining direct associations between different events might not be trivial. However, in order to utilize a machine learning model there are few certain steps to be taken, including labeled data collection, preprocessing, feature extraction and selection, model training, evaluation, and inference.

- **Labeled Data Collection:** The model should be generalizable, which means that it requires a large variety of training dataset covering a wide range of distributions. Therefore, a machine learning algorithms can be a candidate for DNS data analysis only if there is enough labeled dataset. On the other hand, it should be noted that data collection and labeling is a time-consuming process. For instance, for malicious domain detection it is mandatory to have a sufficient number of records of both legitimate and malicious domains.
- **Preprocessing:** Usually the collected dataset is tempered by a wide variety of factors and may include invalid samples, outliers, noises, etc. which also requires a further step to clean the dataset. Generally, a dataset with huge number of invalid samples is of low importance in the machine learning domain.
- **Feature Extraction/Selection:** Once the dataset is obtained and the preprocessing phase is done, it is time to get useful features from the dataset. These features are the inputs of the machine learning model, so they determine the characteristics of the dataset. For example, representative features of IPs can be presented using the number of fully qualified domain names, the number of second level domains, etc. [55]. As it can be understood, this step is highly dependent on the domain knowledge and poor knowledge about the system would highly result in poor features. While the feature extraction aims to get features from the existing dataset, the feature selection method ranks the extracted features based on their importance [160].
- **Model Training, Evaluation, and Inference:** In addition to the previous steps, selecting a proper model structure, and its parameters plays a critical role on the performance of the model. Selecting a proper model

Table 11

Summary of the data collection methods used in research works. Here TB is Terra Byte, M is Million, B is Billion, and LDNS donates local DNS servers. ⁽¹⁾WM and ⁽²⁾ are the Collage of William and Mary and the University of Delaware, respectively.

Work	Method	Volume	Query types												Data scale	Source
			A	AAAA	MX	NS	TXT	PTR	SRV	SOA	CNAME	DS	DNSKEY	NSEC		
[132]	PDNS	1.7TB	✓	✓	✓	✓									2015-01 to 2016-08	25 snapshots of IPv4 list
[133]	PDNS	200M	✓	✓				✓							2011-01 to 2012-03	90 home residential network
[134]	ADNS	Source list	✓	✓	✓	✓	✓			✓	✓				Daily since 2016	com/name/net/org/biz/Alexa 1M
[42]	PDNS	13.1B	✓	✓	✓	✓	✓			✓	✓				2011-01 to 2015-10	Largest ISP in the U.S.
	ADNS	455B	✓	✓	✓	✓	✓			✓	✓				2015-10 to 2016-08	com/name/net/org/biz/Alexa 1M
[23]	PDNS	1.01B	✓	✓		✓	✓	✓	✓	✓					2015-06 to 2015-07	LDNS at WM ⁽¹⁾
	PDNS	0.19B	✓	✓		✓	✓	✓	✓	✓					2015-12	LDNS at UD ⁽²⁾
[52]	PDNS	26B	✓	✓	✓			✓								600 RDNS
[135]	ADNS	Source list	✓	✓	✓	✓	✓			✓	✓	✓	✓	✓	Daily since 2015	com/net/org/info/mobi/name/biz/asia/aero

structure highly depends on the nature of the dataset and the users purpose. For instance, if the collected dataset is a mixture of numerical and categorical features random forest model would work well. On the other hand, if the classification task is a multitask problem the random forest algorithm would intrinsically suite, unlike SVM models, which are intrinsically designed for two-class problems. Consequently, a multitask classification problem should be divided into multiple binary classification problem to use SVM model. On the other hand, if the number of training samples are small the SVM models lead to more robust results in comparison to NN [161]. Furthermore, the training speed of the model is also an important factor to be considered. For instance, the SVM algorithm's fit time complexity is more than quadratic with the number of samples which makes it hard to scale to datasets beyond 10 000 data points [162]. Moreover, in order to evaluate the generalizability of the model and assess how well it is applicable on independent dataset it is required to perform cross-validation assessment. One of the well-known cross-validation methods is K-fold validation, where dataset is divided into k folds and K different models are trained such that each time one of the folds are used as evaluation dataset while the rest is used for model training [163].

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. [55] have designed an association-based scheme for detection of malicious domains with high accuracy and coverage. Furthermore, Gao et al. [52] 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. [60] have utilized statistical measures such as Kullback–Leibler divergence, Jaccard index, and Levenshtein for domain-flux botnet detection. Gomez et al. [139] have studied the application of visualization for understanding the DNS-based network threat analysis.

Discussion and Open Directions. 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.

Table 12

List of common query types and their description.

Query type	Definition
A	IPv4 address
AAAA	IPv6 address
MX	Mail exchanger record
NS	Authoritative name server
TXT	Arbitrary text strings
PTR	Pointer (IP address/hostname)
SRV	Service (service/hostname)
SOA	Start of Authority
CNAME	Canonical Name (Alias/canonical)
DS	Delegation of Signing
DNSKEY	DNSSEC public key
NSEC	Next SECure (No record/two points)

Table 13

Summary of the machine learning methods used in the literature. Here AR is Accuracy Rate, while FPR represents False Positive Rate. The abbreviations of ML algorithms are described in Table 1.

Work	Application	ML algorithm						AR	FPR
		DT	RF	NB	KNN	SVM	MLP		
[132]	Malicious network	✓				✓		97.1%	1.6%
[23]	Cache poisoning	✓	✓					88.0%	1.0%
[55]	Malicious domain	✓	✓			✓		99.0%	1.0%
[18]	Cache poisoning	✓		✓	✓	✓	✓	91.9%	0.6%
[77]	Parked domains		✓					98.7%	0.5%
[159]	Phishing					✓		95.5%	3.5%

Based on our exploration of the literature, we believe there is a significant need into further automation with machine learning, not only for labeling, but also for the appropriate representation of features fed into machine learning algorithms through abstract structures (e.g., dependency representations, such as graphs) and deep neural networks (e.g., serving as high-quality data extractors).

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 [19,30,36,41,99,140,141] and DNSSEC [10,24,118,136,142–146]. This will help us to understand the advantages, e.g., improved security and disadvantages, e.g., overheads of the deployment of the DNSSEC.

Table 14

Topical classification of the DNS research methods addressed in the literature, with sample work.

Work	PDNS	ADNS	Analysis	Scope	Work	PDNS	ADNS	Analysis	Scope
[132]	✓		✓		[116]		✓	✓	
[48]	✓				[117]			✓	
[133]	✓				[77]			✓	
[114]	✓				[142]				✓
[134]	✓	✓			[145]				✓
[13]	✓				[118]				✓
[12]	✓				[24]				✓
[49]	✓				[10]				✓
[42]	✓	✓			[146]				✓
[68]	✓	✓			[20]				✓
[23]	✓		✓		[17]				✓
[52]	✓				[22]	✓			✓
[135]		✓			[50]			✓	
[55]		✓	✓		[80]				✓
[136]	✓	✓		✓	[100]		✓		
[18]			✓		[120]		✓		
[47]			✓		[124]		✓		
[54]			✓		[164]	✓			
[56]			✓		[165]				✓
[137]			✓						

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. [36] have demonstrated that user-side DNS infrastructure is vulnerable to record injection threats. Moreover, Xu et al. [30] have demonstrated that attackers can severely abuse DNS-based stealthy C&C channel to efficiently hide malicious DNS activities. In addition, Schomp et al. [141] have showed that shared DNS resolvers are vulnerable to various forms of attack [36, 118], 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 [24,146].

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 Fig. 13.

Chung et al. [136] 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. [166] 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. [142], have conducted a measurement

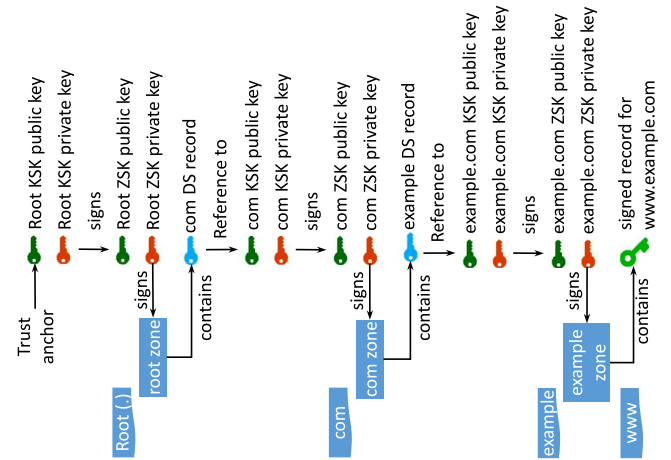


Fig. 13. An overview of DNSSEC chain of trust. DNSSEC utilizes public and private key pairs to verify the integrity and authenticity of DNS records based on a chain-of-trust. The verification process of DNS record begins at the root and continues until the record in question is authenticated.

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. [167], and in general, DNSSEC misconfigurations can be broadly classified into four categories: DNSKEY, RRSIG, general DNS failure, and miscellaneous. Fig. 14 shows the categories and the subcategories of DNSSEC misconfigurations [142,167].

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. [145] 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 and Open Directions. 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 [136]. 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 [118]. 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, and understanding those issues, in the abstract and in context (through updated measurements) is a constant research need that will help significantly improve the security landscape of DNS.

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 Fig. 15, and in Table 16, we present the topical classification of the DNS entities as addressed in the literature, with sample works.

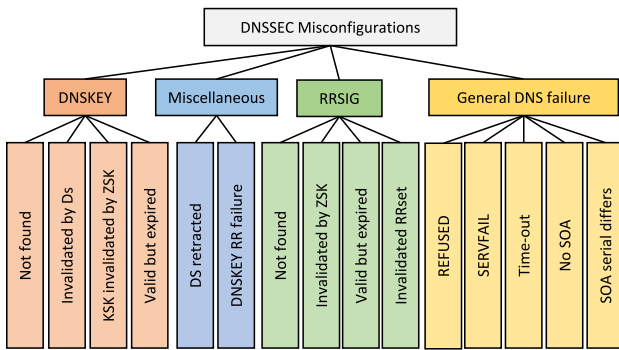


Fig. 14. Overview of DNSSEC misconfigurations. In general, DNSSEC misconfigurations can be categorized in four groups, including DNSKEY, RRSIG, general DNS failure, and miscellaneous.

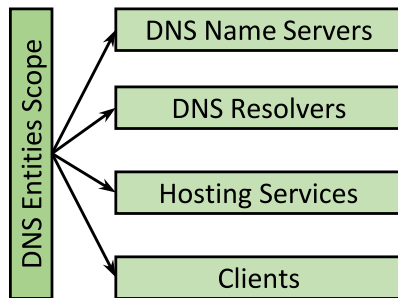


Fig. 15. DNS Entities Scope. We have categorized research works based on the main scope of the entities into DNS name servers, open DNS resolvers, hosting services, and clients.

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 [17,48,52,73,168–173].

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. [168] 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. [172] 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. [51] 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. [173] presented

Name Server Switching Fingerprint (NSSF), a feature modality and a system that analyze 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. [73] have investigated the security practices of authoritative name servers, and observed that a large number of domain names ($\approx 1.28M$) are under the risk of DoS attacks due to weak security practices of few (≈ 7214) authoritative name servers. They also suggested employment of DNSSEC as a solution to enhance the security of DNS. Furthermore, Chung et al. [136] 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. [14,22,23,48,169]. Chen et al. [22] 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 RRs, that are unlikely to be reused. Hao and Wang [23] 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 RRs into the cache can prevent a waste of the DNS cache resources.

Shang et al. [170] explored the relationships across domain names to improve the cache hit rate of local DNS servers. In addition, Chen et al. [169] 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. [20] have proposed a novel method based on WSEC DNS to mitigate recursive DNS resolvers' cache poisoning attack.

Ager et al. [174] 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 [175] and OpenDNS [176]. They further observed that unlike third-party DNS resolvers, local DNS resolvers redirect clients towards the content available within the ISP. Kuhrer et al. [37] 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. [14] 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. [8] have analyzed the caching behavior of DNS resolvers to defend against DNS DoS attacks.

Discussion and Open Directions. Although top level DNS servers perform well, the quality of service is unfair around the globe [172]. 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 [73]. 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 [136]. All in all, the state-of-the-art indicates a gap in the governance and compliance of the deployment in essential name servers, making their security through measurements, analysis, and defenses of paramount importance.

Table 15

Summary of the research works that measured the number of open DNS resolvers.

Work	Year	Scope	# ORDNS
[177]	2010	105.4M IPs	11.9M
	2010	1.1M IPs	114.7K
[178]	2013	IPv4 IPs	32M
[179,180]	2013	IPv4 IPs	32M
	2017	IPv4 IPs	10.3M
[181]	2013	IPv4 IPs	11.9M
	2017	IPv4 IPs	3.7M
[182]	2013	IPv4 IPs	33.6M
	2014	IPv4 IPs	29.2M
[183]	2013	IPv4 IPs	25M

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 resolver 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 Figs. 16 and 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 [8,14,18,20,24,36–38,52,78,141,174].

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 [177] 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. [18] 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. [28] 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. [178] 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. [52] 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. [184] 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 min. The Open Resolver Project [179,180] 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 [181] 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. [183] performed measurements based on responses for

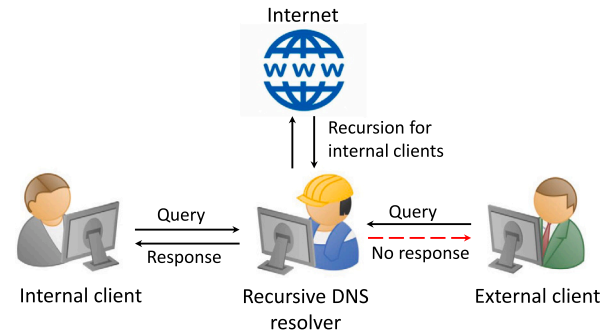


Fig. 16. Overview of recursive DNS resolver. Notice that RDNS resolver does not respond to queries of external clients.

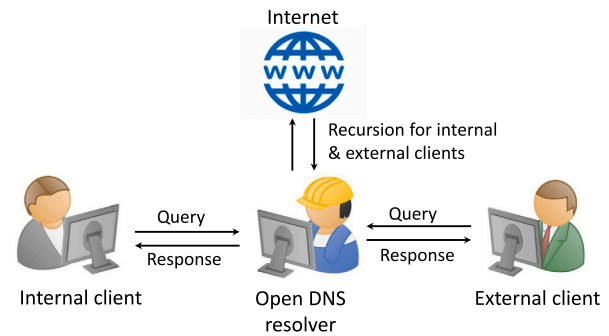


Fig. 17. Overview of open recursive DNS resolver. Notice that ORDNS resolver does respond to queries of both internal and external clients.

Internet-wide DNS software version requests. They have focused on DNS server software and their distribution in each regional Internet registry.

In a more recent work, Park et al. [185] have conducted a behavioral analysis of the open DNS resolvers. They have scanned the whole IPv4 space and found that more than 3 million open resolvers still exist in the wild, among which many of them answer queries with the incorrect, even malicious, responses.

5.2.3. Threats

Antonakakis et al. [18] noticed that in general, the attackers point victims to rogue IP addresses through open DNS resolvers. Kuhrer et al. [37] 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. [36] 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. [14] 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. [8] 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. [78] have studied third party DNS resolvers and consequent threats to online privacy. They proposed a new privacy-preserving name resolution service

Table 16

Topical classification of the DNS research scopes addressed in the literature, with sample work; NS=Name Servers, R=Resolvers, HS=Hosting Services, and C=Clients.

Work	NS	R	HS	C	Work	NS	R	HS	C
[142]			✓		[172]	✓			
[174]	✓	✓			[56]				✓
[18]		✓			[57]			✓	
[48]	✓				[187]				✓
[8]		✓			[38]		✓		
[169]	✓				[20]		✓		
[136]	✓		✓		[188]		✓		
[52]		✓			[141]		✓		
[189]		✓			[36]		✓		✓
[23]	✓				[190]				✓
[168]	✓		✓		[170]	✓			
[78]		✓			[17]	✓			
[118]	✓	✓			[117]		✓		
[24]		✓			[114]				✓
[10]			✓		[13]	✓			
[28]		✓			[14]		✓		
[68]				✓	[73]	✓			
[37]		✓		✓	[126]		✓		
[115]			✓						

that relies on the encapsulation of encrypted messages in standards-compliant DNS messages. Hendriks et al. [186] 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. [141] have studied the vulnerabilities of shared DNS resolvers and proposed an unorthodox method to tackle such threats.

Discussion and Open Directions. 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 [172]. 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 [142], 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% .org properly signed domains [136]. Chung et al. [136] 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. [168] 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 life-cycle of servers. They have heuristically analyzed the Alexa's top 1-million 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. [10] have studied DNS-amplification DoS attacks, and have proposed a defense system that is compatible with common DNS servers configurations and DNSSEC. They 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 and Open Directions. Cloud services are increasingly popular due to their affordability [168]. 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, especially where there is a dependency in its security on other critical infrastructure; e.g., DNS.

5.4. Clients

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 [36,37,56,68,187,189,190]. For example, Schomp et al. [190] 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. [187] 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. [189] 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. [37] 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. [68] 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. [56] 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 and Open Directions. 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.

CRedit authorship contribution statement

Aminollah Khormali: Initiated the study, Compiled the list of works, Written the majority of the work, Revised the submission. **Jeman Park:** Summarizing some of the works in the survey, and discussing their shortcomings. **Hisham Alasmay:** Summarizing some of the works in the survey, and discussing their shortcomings. **Afsah Anwar:** Summarizing some of the works in the survey, and discussing their shortcomings. **Muhammad Saad:** Summarizing some of the works in the initial survey, and discussing their shortcomings; helped with the final revision. **David Mohaisen:** Designed the various survey aspects, Written the majority of the work, Written the discussion and open directions, Revised the submission.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

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

References

- [1] F. Zou, S. Zhang, B. Pei, L. Pan, L. Li, J. Li, Survey on domain name system security, in: Proceedings of the IEEE First International Conference on Data Science in Cyberspace DSC, 2016, pp. 602–607.
- [2] A.R. Kang, J. Spaulding, A. Mohaisen, Domain name system security and privacy: Old problems and new challenges, CoRR abs/1606.07080.
- [3] S. Torabi, A. Boukhtouta, C. Assi, M. Debbabi, Detecting internet abuse by analyzing passive dns traffic: A survey of implemented systems, IEEE Commun. Surv. Tutor.
- [4] Y. Zhauniarovich, I. Khalil, T. Yu, M. Dacier, A survey on malicious domains detection through dns data analysis, ACM Comput. Surv. 51 (4).
- [5] J. Spaulding, S.J. Upadhyaya, A. Mohaisen, The landscape of domain name typosquatting: Techniques and countermeasures, in: Proceedings of the 11th International Conference on Availability, Reliability and Security ARES, 2016, pp. 284–289.
- [6] P.V. Mockapetris, Domain names: Implementation specification, RFC 883 (1983) 1–74.
- [7] A.A. Aizuddin, M. Atan, M. Norulazmi, M.M. Noor, S. Akimi, Z. 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, 2017, p. 3.
- [8] H. Ballani, P. Francis, A simple approach to dns dos mitigation, Microsoft.
- [9] S. Kim, S. Lee, G. Cho, M.E. Ahmed, J.P. Jeong, H. 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), Oslo, Norway, 2017, pp. 135–152.
- [10] A. Herzberg, H. Shulman, DNS authentication as a service: preventing amplification attacks, in: Proceedings of the 30th Annual Computer Security Applications Conference ACSAC, 2014, pp. 356–365.
- [11] D.C. MacFarland, C.A. Shue, A.J. Kalafut, Characterizing optimal DNS amplification attacks and effective mitigation, in: Proceedings of the 16th International Conference on Passive and Active Measurement PAM, New York, NY, 2015, pp. 15–27.
- [12] R. Perdisci, I. Corona, G. Giacinto, Early detection of malicious flux networks via large-scale passive DNS traffic analysis, IEEE Trans. Dependable Secure Comput. 9 (5) (2012) 714–726.
- [13] D. Truong, G. Cheng, Detecting domain-flux botnet based on DNS traffic features in managed network, Secur. Commun. Netw. 9 (14) (2016) 2338–2347.
- [14] S. Verma, A. Hamieh, J.H. Huh, H. Holm, S.R. Rajagopalan, M. Korczynski, N.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, 2016, pp. 69–78.
- [15] J. Zheng, Q. Li, G. Gu, J. Cao, D.K. Yau, J. Wu, Realtime ddos defense using cots sdn switches via adaptive correlation analysis, IEEE Trans. Inf. Forensics Secur. 13 (7) (2018) 1838–1853.
- [16] N. Limited, Ddos threats report 2018 q1, <https://www.nexusguard.com/threat-report-q1-2018>.
- [17] H. Shulman, M. Waidner, Towards forensic analysis of attacks with DNSSEC, in: Proceedings of the 35th IEEE Symposium on Security and Privacy (Oakland), San Jose, CA, 2014, pp. 69–76.
- [18] M. Antonakakis, D. Dagon, X. Luo, R. Perdisci, W. Lee, J. Bellmor, A centralized monitoring infrastructure for improving DNS security, in: Proceedings of the 15th International Symposium on Research in Attacks, Intrusions and Defenses (RAID), Ottawa, Canada, 2010, pp. 18–37.
- [19] C. Chen, S. Matsumoto, A. Perrig, ECO-DNS: expected consistency optimization for DNS, in: Proceedings of the 35th International Conference on Distributed Computing Systems (ICDCS), 2015, pp. 256–267.
- [20] R. Perdisci, M. Antonakakis, X. Luo, W. Lee, WSEC DNS: protecting recursive DNS resolvers from poisoning attacks, in: Proceedings of the 39th International Conference on Dependable Systems and Networks (DSN), Lisbon, Portugal, 2009, pp. 3–12.
- [21] H. Wu, X. Dang, L. Zhang, L. Wang, Kalman filter based DNS cache poisoning attack detection, in: Proceedings of the IEEE International Conference on Automation Science and Engineering CASE, 2015, pp. 1594–1600.
- [22] Y. Chen, M. Antonakakis, R. Perdisci, Y. Nadji, D. Dagon, W. Lee, DNS noise: Measuring the pervasiveness of disposable domains in modern DNS traffic, in: Proceedings of the 44th Annual IEEE/IFIP International Conference on Dependable Systems and Networks DSN, 2014, pp. 598–609.
- [23] S. Hao, H. Wang, Exploring domain name based features on the effectiveness of DNS caching, Comput. Commun. Rev. 47 (1) (2017) 36–42.
- [24] A. Herzberg, H. Shulman, Security of patched DNS, in: Proceedings of the 17th European Symposium on Research in Computer Security (ESORICS), Pisa, Italy, 2012, pp. 271–288.
- [25] Rustock ddos attack [cited 12.02.2018], <http://www.joestewart.org/rustock-ddos.html>.
- [26] M. Riccardi, R. Di Pietro, M. Palanques, J.A. Vila, Titans' revenge: Detecting zeus via its own flaws, Comput. Netw. 57 (2) (2013) 422–435.

- [27] J. Jin, Z. Yan, G. Geng, B. Yan, Botnet domain name detection based on machine learning, in: Proceedings of the 6th International Conference on Wireless, Mobile and Multi-Media ICWMMN, 2015.
- [28] J. Jiang, J. Liang, K. Li, J. Li, H. Duan, J. Wu, Ghost domain names: Revoked yet still resolvable, in: Proceedings of the 19th Annual Network and Distributed System Security Symposium (NDSS), San Diego, CA, 2012.
- [29] D. Plohmann, K. Yakdan, M. Klatt, J. Bader, E. Gerhards-Padilla, A comprehensive measurement study of domain generating malware, in: Proceedings of the 25th USENIX Security Symposium (Security), Austin, TX, 2016, pp. 263–278.
- [30] K. Xu, P. Butler, S. Saha, D.D. Yao, DNS for massive-scale command and control, *IEEE Trans. Dependable Secur. Comput.* 10 (3) (2013) 143–153.
- [31] Z. Whittaker, Mirai botnet attackers are trying to knock an entire country offline (2016) [cited 12.02.2018]. <https://www.zdnet.com/article/mirai-botnet-attack-briefly-knocked-an-entire-country-offline/>.
- [32] D. Palmer, Mirai botnet adds three new attacks to target iot devices (2018) [cited 12.02.2018], <https://www.zdnet.com/article/mirai-botnet-adds-three-new-attacks-to-target-iot-devices/>.
- [33] C. Kolias, G. Kambourakis, A. Stavrou, J. Voas, Ddos in the iot: Mirai and other botnets, *Computer* 50 (7) (2017) 80–84.
- [34] E. Bertino, N. Islam, Botnets and internet of things security, *Computer* 50 (2) (2017) 76–79.
- [35] N. Weaver, C. Kreibich, B. Nechaev, V. Paxson, Implications of netalizer dns measurements, in: Proceedings of the First Workshop on Securing and Trusting Internet Names, Citeseer, 2011.
- [36] K. Schomp, T. Callahan, M. Rabinovich, M. Allman, Assessing DNS vulnerability to record injection, in: Proceedings of the 15th International Conference on Passive and Active Measurement PAM, Los Angeles, CA, 2014, pp. 214–223.
- [37] M. Kühner, T. Hüpperich, J. Bushart, C. Rossow, T. Holz, Going wild: Large-scale classification of open DNS resolvers, in: Proceedings of the 2015 Internet Measurement Conference (IMC), Tokyo, Japan, 2015, pp. 355–368.
- [38] P. Pearce, B. Jones, F. Li, R. Ensafi, N. Feamster, N. Weaver, V. Paxson, Global measurement of DNS manipulation, in: Proceedings of the 26th USENIX Security Symposium (Security), Vancouver, BC, Canada, 2017, pp. 307–323.
- [39] M. Trevisan, I. Drago, M. Mellia, M.M. Munafò, Automatic detection of DNS manipulations, in: Proceedings of the IEEE International Conference on Big Data, 2017, pp. 4010–4015.
- [40] L. Bilge, E. Kirda, C. Kruegel, M. 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, 2011.
- [41] D. Dagon, N. Provos, C.P. Lee, W. Lee, Corrupted DNS resolution paths: The rise of a malicious resolution authority, in: Proceedings of the 15th Annual Network and Distributed System Security Symposium (NDSS), San Diego, CA, 2008.
- [42] P. Kintis, N. Miramirkhani, C. Lever, Y. Chen, R.R. Gómez, N. Pitropakis, N. Nikiforakis, M. Antonakakis, Hiding in plain sight: A longitudinal study of combsquatting abuse, in: Proceedings of the 24th ACM Conference on Computer and Communications Security (CCS), Dallas, TX, 2017, pp. 569–586.
- [43] D.L. Cook, V.K. Gurbani, M. Daniluk, Phishwish: A stateless phishing filter using minimal rules, in: Proceedings of the 12th International Conference on Financial Cryptography and Data Security FC, 2008, pp. 182–186.
- [44] E. Medvet, E. Kirda, C. Kruegel, Visual-similarity-based phishing detection, in: proceedings of the 4th International ICST Conference on Security and Privacy in Communication Networks SECURECOMM, 2008, p. 22.
- [45] M. Hara, A. Yamada, Y. Miyake, Visual similarity-based phishing detection without victim site information, in: Proceedings of the 2009 IEEE Symposium on Computational Intelligence in Cyber Security, CICS, 2009, pp. 30–36.
- [46] . Wandera, Wandera report finds phishing attacks are moving to mobile messaging and social apps at an alarming rate (2018) [cited 12.02.2018], <https://bit.ly/2Ky11pd>.
- [47] M. Antonakakis, R. Perdisci, D. Dagon, W. Lee, N. Feamster, Building a dynamic reputation system for DNS, in: Proceedings of the 19th USENIX Security Symposium (Security), Washington, DC, 2010, pp. 273–290.
- [48] M. Antonakakis, R. Perdisci, W. Lee, N. Vasiloglou, D. Dagon, Detecting malware domains at the upper DNS hierarchy, in: Proceedings of the 20th USENIX Security Symposium (Security), San Francisco, CA, 2011.
- [49] L. Bilge, S. Sen, D. Balzarotti, E. Kirda, C. Kruegel, Exposure: A passive DNS analysis service to detect and report malicious domains, *ACM Trans. Inf. Syst.* 16 (4) (2014) 14:1–14:28.
- [50] D. Canali, M. Cova, G. Vigna, C. Kruegel, Proceedings of the 20th international world wide web conference (www), in: Proceedings of the 20th International World Wide Web Conference (WWW), Hyderabad, India, 2011, pp. 197–206.
- [51] M. Félegyházi, C. Kreibich, V. Paxson, On the potential of proactive domain blacklisting, in: Proceedings of the 3rd USENIX Workshop on Large-Scale Exploits and Emergent Threats, 2010.
- [52] H. Gao, V. Yegneswaran, Y. Chen, P.A. Porras, S. Ghosh, J. Jiang, H. Duan, An empirical reexamination of global DNS behavior, in: Proceedings of the 2013 ACM SIGCOMM, Hong Kong, China, 2013, pp. 267–278.
- [53] S. Hao, A. Kantchelian, B. Miller, V. Paxson, N. 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), Vienna, Austria, 2016, pp. 1568–1579.
- [54] S. Hao, M. Thomas, V. Paxson, N. Feamster, C. Kreibich, C. Grier, S. Hollenbeck, Understanding the domain registration behavior of spammers, in: Proceedings of the Internet Measurement Conference (IMC), Barcelona, Spain, 2013, pp. 63–76.
- [55] I. Khalil, B. Guan, M. Nabeel, T. 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.
- [56] P. Luo, R. Torres, Z. Zhang, S. Saha, S. Lee, A. Nucci, M. Mellia, Leveraging client-side DNS failure patterns to identify malicious behaviors, in: Proceedings of the IEEE Conference on Communications and Network Security CNS, 2015, pp. 406–414.
- [57] J. Ma, L.K. Saul, S. Savage, G.M. Voelker, Identifying suspicious urls: an application of large-scale online learning, in: Proceedings of the 26th International Conference on Machine Learning (ICML), Montreal, Canada, 2009, pp. 681–688.
- [58] S. Hao, N. Feamster, R. Pandrangi, Monitoring the initial DNS behavior of malicious domains, in: Proceedings of the Internet Measurement Conference (IMC), Berlin, Germany, 2011, pp. 269–278.
- [59] D. Liu, Z. Li, K. Du, H. Wang, B. Liu, H. 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), Dallas, TX, 2017, pp. 537–552.
- [60] S. Yadav, A.K.K. Reddy, A.L.N. Reddy, S. Ranjan, Detecting algorithmically generated malicious domain names, in: Proceedings of the Internet Measurement Conference (IMC), Melbourne, Australia, 2010, pp. 48–61.
- [61] M. Antonakakis, R. Perdisci, Y. Nadj, N. Vasiloglou, S. Abu-Nimeh, W. Lee, D. Dagon, From throw-away traffic to bots: Detecting the rise of dga-based malware, in: Proceedings of the 21st USENIX Security Symposium (Security), 12, Bellevue, WA, 2012.
- [62] T. Barabosch, A. Wichmann, F. Leder, E. 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.
- [63] J. Spaulding, J. Park, J. Kim, A. Mohaisen, Proactive detection of algorithmically generated malicious domains, in: Proceedings of the International Conference on Information Networking, ICOIN, 2018, pp. 21–24.
- [64] Y. Wang, D. Beck, J. Wang, C. Verbowski, B. 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, SRUT'06, 2006.
- [65] A. Banerjee, D. Barman, M. Faloutsos, L.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, 2008, pp. 1939–1947.
- [66] P. Ageton, W. Joosen, F. Piessens, N. 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, 2015.
- [67] J. Szurdi, B. Kocso, G. Cseh, J. Spring, M. Félegyházi, C. Kanich, The long tale of typosquatting domain names, in: Proceedings of the 23rd USENIX Security Symposium (Security), San Diego, CA, 2014, pp. 191–206.
- [68] M.T. Khan, X. Huo, Z. Li, C. Kanich, Every second counts: Quantifying the negative externalities of cybercrime via typosquatting, in: Proceedings of the 36th IEEE Symposium on Security and Privacy (Oakland), San Jose, CA, 2015, pp. 135–150.
- [69] J. Spaulding, D. Nyang, A. Mohaisen, Understanding the effectiveness of typosquatting techniques, in: Proceedings of the fifth ACM/IEEE Workshop on Hot Topics in Web Systems and Technologies, HotWeb, 2017, pp. 9:1–9:8.
- [70] J. Spaulding, S.J. Upadhyaya, A. Mohaisen, You've been tricked! A user study of the effectiveness of typosquatting techniques, in: Proceedings of the 37th International Conference on Distributed Computing Systems (ICDCS), 2017, pp. 2593–2596.
- [71] Bitsquatting: Dns hijacking without exploitation [cited 12.02.2018], <http://dinaburg.org/bitsquatting.html>.
- [72] N. Nikiforakis, S.V. Acker, W. Meert, L. Desmet, F. Piessens, W. Joosen, Bitsquatting: exploiting bit-flips for fun, or profit?, in: Proceedings of the 22nd International World Wide Web Conference (WWW), 2013, pp. 989–998.
- [73] T. Vissers, T. Barron, T. van Goethem, W. Joosen, N. Nikiforakis, The wolf of name street: Hijacking domains through their nameservers, in: Proceedings of the 24th ACM Conference on Computer and Communications Security (CCS), Dallas, TX, 2017, pp. 957–970.
- [74] N. Nikiforakis, M. Balduzzi, L. Desmet, F. Piessens, W. Joosen, Soundsquatting: Uncovering the use of homophones in domain squatting, in: Proceedings of the 17th Information Security Conference (ISC), Hong Kong, 2014, pp. 291–308.
- [75] S.A. Alrwais, K. Yuan, E. Alowaisheq, Z. Li, X. Wang, Understanding the dark side of domain parking, in: Proceedings of the 23rd USENIX Security Symposium (Security), San Diego, CA, 2014, pp. 207–222.

- [76] N. Weaver, C. Kreibich, V. Paxson, Redirecting dns for ads and profit, *FOCI 2* (2011) 2–3.
- [77] T. Vissers, W. Joosen, N. Nikiforakis, Parking sensors: Analyzing and detecting parked domains, in: *Proceedings of the 2015 Annual Network and Distributed System Security Symposium (NDSS)*, San Diego, CA, 2015.
- [78] D. Herrmann, K. Fuchs, J. Lindemann, H. Federrath, Encdns: A lightweight privacy-preserving name resolution service, in: *Proceedings of the 19th European Symposium on Research in Computer Security (ESORICS)*, Wroclaw, Poland, 2014, pp. 37–55.
- [79] Y. Lu, G. Tsudik, Towards plugging privacy leaks in the domain name system, in: *Proceedings of the 10th IEEE International Conference on Peer-to-Peer Computing P2P*, 2010, pp. 1–10.
- [80] A. Mohaisen, Z. Gu, K. Ren, Privacy implications of DNSSEC look-aside validation, in: *Proceedings of the 37st International Conference on Distributed Computing Systems (ICDCS)*, 2017, pp. 2438–2443.
- [81] C. Rossow, Amplification hell: Revisiting network protocols for ddos abuse, in: *Proceedings of the 21st Annual Network and Distributed System Security Symposium*, 2014.
- [82] R. van Rijswijk-Deij, A. Sperotto, A. Pras, DNSSEC and its potential for ddos attacks: a comprehensive measurement study, in: *Proceedings of the 2014 Internet Measurement Conference, IMC*, 2014, pp. 449–460.
- [83] D. Erhan, E. Anarim, Hybrid ddos detection framework using matching pursuit algorithm, *IEEE Access* 8 (2020) 118912–118923.
- [84] W. Li, K. Shen, R. Guo, B. Liu, J. Zhang, H. Duan, S. Hao, X. Chen, Y. Wang, Cdn backfired: Amplification attacks based on http range requests, in: *2020 50th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, IEEE, 2020, pp. 14–25.
- [85] S.-Y. Chang, Y. Park, N.V. Kengalahalli, X. Zhou, Query-crafting dos threats against internet dns, in: *2020 IEEE Conference on Communications and Network Security (CNS)*, IEEE, 2020, pp. 1–9.
- [86] C. Kreibich, N. Weaver, B. Nechaev, V. Paxson, Netalyzr: illuminating the edge network, in: *Proceedings of the Internet Measurement Conference (IMC)*, Melbourne, Australia, 2010, pp. 246–259.
- [87] M. Larsen, F. Gont, Recommendations for transport-protocol port randomization (1970) [cited 12.02.2018]. <http://www.rfc-editor.org/info/rfc6056>.
- [88] A. Hubert, R.v. Mook, Measures for making dns more resilient against forged answers (1970) [cited 12.02.2018], <https://www.rfc-editor.org/info/rfc5452>.
- [89] F. Alharbi, J. Chang, Y. Zhou, F. Qian, Z. Qian, N. Abu-Ghazaleh, Collaborative client-side dns cache poisoning attack, in: *In the Proceedings of the IEEE Conference on Computer Communications, INFOCOM 2019, IEEE, 2019, pp. 1153–1161*.
- [90] X. Zheng, C. Lu, J. Peng, Q. Yang, D. Zhou, B. Liu, K. Man, S. Hao, H. Duan, Z. Qian, Poison over troubled forwarders: A cache poisoning attack targeting {DNS} forwarding devices, in: *29th {USENIX} Security Symposium ({USENIX} Security 20)*, 2020, pp. 577–593.
- [91] W. Chang, A. Mohaisen, A. Wang, S. Chen, Understanding adversarial strategies from bot recruitment to scheduling, in: *Proceedings of the 13th International Conference on Security and Privacy in Communication Networks -, SecureComm 2017*, 2017, pp. 397–417.
- [92] M. Thomas, A. Mohaisen, Kindred domains: detecting and clustering botnet domains using DNS traffic, in: *Proceedings of the 23th International World Wide Web Conference (WWW)*, 2014, pp. 707–712.
- [93] A. Mohaisen, O. Alrawi, Unveiling zeus: automated classification of malware samples, in: *Proceedings of the 22nd International World Wide Web Conference (WWW)*, 2013, pp. 829–832.
- [94] R. Vinayakumar, M. Alazab, S. Srinivasan, Q.-V. Pham, S.K. Padannayil, K. Simran, A visualized botnet detection system based deep learning for the internet of things networks of smart cities, *IEEE Trans. Ind. Appl.*
- [95] W. Wang, Y. Shang, Y. He, Y. Li, J. Liu, Botmark: Automated botnet detection with hybrid analysis of flow-based and graph-based traffic behaviors, *Inform. Sci.* 511 (2020) 284–296.
- [96] M. Singapore, Microsoft consumer safety index reveals impact of poor online safety behaviours in singapore [cited 06.02.2018], <https://bit.ly/2y7h9B5>.
- [97] A. Oest, P. Zhang, B. Wardman, E. Nunes, J. Burgis, A. Zand, K. Thomas, A. Doupe, G.-J. Ahn, Sunrise to sunset: Analyzing the end-to-end life cycle and effectiveness of phishing attacks at scale, in: *29th {USENIX} Security Symposium ({USENIX} Security 20)*, 2020.
- [98] D. Kaminsky, Black ops 2008: It's the end of the cache as we know it, *Black Hat USA*.
- [99] D. Dagon, M. Antonakakis, P. Vixie, T. Jinmei, W. Lee, Increased DNS forgery resistance through 0x20-bit encoding: security via leet queries, in: *Proceedings of the 15th ACM Conference on Computer and Communications Security (CCS)*, Alexandria, VA, 2008, pp. 211–222.
- [100] B. Jones, N. Feamster, V. Paxson, N. Weaver, M. Allman, Detecting DNS root manipulation, in: *Proceedings of the 17th International Conference on Passive and Active Measurement PAM*, Heraklion, Greece, 2016, pp. 276–288.
- [101] X. Sun, M. Tong, J. Yang, L. Xinran, L. Heng, Hindom: A robust malicious domain detection system based on heterogeneous information network with transductive classification, in: *22nd International Symposium on Research in Attacks, Intrusions and Defenses ({RAID} 2019)*, 2019, pp. 399–412.
- [102] X. Sun, Z. Wang, J. Yang, X. Liu, Deepdom: Malicious domain detection with scalable and heterogeneous graph convolutional networks, *Comput. Secur.* (2020) 102057.
- [103] A.O. Almahshdani, M. Kaiiali, D. Carlin, S. Sezer, Maldomdetector: A system for detecting algorithmically generated domain names with machine learning, *Comput. Secur.* (2020) 101787.
- [104] C. Xu, J. Shen, X. Du, Detection method of domain names generated by dgas based on semantic representation and deep neural network, *Comput. Secur.* 85 (2019) 77–88.
- [105] B. Yu, J. Pan, D. Gray, J. Hu, C. Choudhary, A.C. Nascimento, M. De Cock, Weakly supervised deep learning for the detection of domain generation algorithms, *IEEE Access* 7 (2019) 51542–51556.
- [106] S. Kullback, R.A. Leibler, On information and sufficiency, *The ann. math. stat.* 22 (1) (1951) 79–86.
- [107] F. Zhao, Y. Hori, K. Sakurai, Analysis of privacy disclosure in DNS query, in: *Proceedings of the 2007 International Conference on Multimedia and Ubiquitous Engineering MUE*, 2007, pp. 952–957.
- [108] F. Zhao, Y. Hori, K. Sakurai, Two-servers PIR based DNS query scheme with privacy-preserving, in: *Proceedings of the 2007 International Conference on Intelligent Pervasive Computing, IPC*, Jeju City, South Korea, 2007, pp. 299–302.
- [109] L. Zhu, Z. Hu, J.S. Heidemann, D. Wessels, A. Mankin, N. Somaiya, Connection-oriented DNS to improve privacy and security, in: *Proceedings of the 36th IEEE Symposium on Security and Privacy (Oakland)*, San Jose, CA, 2015, pp. 171–186.
- [110] S. Krishnan, F. Monroe, 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.
- [111] A. King, Thunder screenreader [cited 12.02.2018], <https://www.webbie.org.uk/thunder/>.
- [112] Welcome to orca [cited 12.02.2018], <https://help.gnome.org/users/orca/stable/introduction.html.en>.
- [113] Y. Hu, H. Wang, R. He, L. Li, G. Tyson, I. Castro, Y. Guo, L. Wu, G. Xu, Mobile app squatting, in: *Proceedings of The Web Conference 2020*, 2020, pp. 1727–1738.
- [114] S. Tajalizadehkhoob, T. van Goethem, M. Korczynski, A. Noroozian, R. Böhme, T. Moore, W. Joosen, M. 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)*, Dallas, TX, 2017, pp. 553–567.
- [115] Z. Li, S.A. Alrwais, Y. Xie, F. Yu, X. 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)*, San Francisco, CA, 2013, pp. 112–126.
- [116] B. Rahbarinia, R. Perdisci, M. Antonakakis, Segugio: 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*, 2015, pp. 403–414.
- [117] H. Shulman, M. Waidner, Towards security of internet naming infrastructure, in: *Proceedings of the 20th European Symposium on Research in Computer Security (ESORICS)*, Vienna, Austria, 2015, pp. 3–22.
- [118] A. Herzberg, H. Shulman, Fragmentation considered poisonous, or: One-domain-to-rule-them-all.org, in: *Proceedings of the IEEE Conference on Communications and Network Security CNS*, 2013, pp. 224–232.
- [119] S. Castillo-Perez, J. García-Alfaro, Evaluation of two privacy-preserving protocols for the DNS, in: *Proceedings of the 6th International Conference on Information Technology: New Generations ITNG*, 2009, pp. 411–416.
- [120] V. Paxson, M. Christodorescu, M. Javed, J.R. Rao, R. Sailer, D.L. Schales, M.P. Stoecklin, K. Thomas, W. Venema, N. Weaver, Practical comprehensive bounds on surreptitious communication over DNS, in: *Proceedings of the 22th USENIX Security Symposium (Security)*, Washington, DC, 2013, pp. 17–32.
- [121] A. Mohaisen, K. Ren, Leakage of onion at the DNS root: Measurements, causes, and countermeasures, *IEEE/IEEE/ACM Trans. Netw.* 25 (5) (2017) 3059–3072.
- [122] A. Mohaisen, A.R. Kang, K. 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*, Jeju Island, Korea, 2016, pp. 1–14.
- [123] B. Benschhof, A. Rosen, A.G. Bourgeois, R.W. Harrison, Distributed decentralized domain name service, in: *Proceedings of the IEEE International Parallel and Distributed Processing Symposium Workshops IPDPS*, 2016, pp. 1279–1287.
- [124] K. Du, H. Yang, Z. Li, H. Duan, K. Zhang, The ever-changing labyrinth: A large-scale analysis of wildcard DNS powered blackhat SEO, in: *Proceedings of the 25th USENIX Security Symposium (Security)*, Austin, TX, 2016, pp. 245–262.

- [125] C. Lever, R.J. Walls, Y. Nadji, D. Dagon, P.D. McDaniel, M. Antonakakis, Domain-z: 28 registrations later measuring the exploitation of residual trust in domains, in: Proceedings of the 37th IEEE Symposium on Security and Privacy (Oakland), San Jose, CA, 2016, pp. 691–706.
- [126] Y. Yu, D. Wessels, M. Larson, L. Zhang, Authority server selection in DNS caching resolvers, *Comput. Commun. Rev.* 42 (2) (2012) 80–86.
- [127] D. Liu, S. Hao, H. 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), Vienna, Austria, 2016, pp. 1414–1425.
- [128] H. Shulman, Pretty bad privacy: Pitfalls of DNS encryption, in: Proceedings of the 13th Workshop on Privacy in the Electronic Society, WPES, 2014, pp. 191–200.
- [129] Dns over dtls (dnsod) [cited 12.02.2018], <https://tools.ietf.org/id/draft-ietf-dprive-dnsodtls-07.html>.
- [130] H. Federrath, K. Fuchs, D. Herrmann, C. Pioscny, Privacy-preserving DNS: analysis of broadcast, range queries and mix-based protection methods, in: Proceedings of the 16th European Symposium on Research in Computer Security (ESORICS), Leuven, Belgium, 2011, pp. 665–683.
- [131] The onion special-use domain name [cited 12.02.2018], <https://tools.ietf.org/html/rfc7686>.
- [132] S.A. Alrwais, X. Liao, X. Mi, P. Wang, X. Wang, F. Qian, R.A. Beyah, D. 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), San Jose, CA, 2017, pp. 805–823.
- [133] T. Callahan, M. Allman, M. Rabinovich, On modern DNS behavior and properties, *Comput. Commun. Rev.* 43 (3) (2013) 7–15.
- [134] A. Kountouras, P. Kintis, C. Lever, Y. Chen, Y. Nadji, D. Dagon, M. Antonakakis, R. Joffe, Enabling network security through active DNS datasets, in: Proceedings of the 19th International Symposium on Research in Attacks, Intrusions and Defenses (RAID), Evry, France, 2016, pp. 188–208.
- [135] R. van Rijswijk-Deij, M. Jonker, A. Sperotto, A. Pras, A. high performance, Scalable infrastructure for large-scale active dns measurements, *IEEE J. Sel. Areas Commun.* 34 (6) (2016) 1877–1888.
- [136] T. Chung, R. van Rijswijk-Deij, B. Chandrasekaran, D.R. Choffnes, D. Levin, B.M. Maggs, A. Mislove, C. Wilson, A. longitudinal, A longitudinal, end-to-end view of the DNSSEC ecosystem, in: Proceedings of the 26th USENIX Security Symposium (Security), Vancouver, BC, Canada, 2017, pp. 1307–1322.
- [137] M. Radwan, R. Heckel, Prediction of the domain name system (dns) quality attributes, in: Proceedings of the Symposium on Applied Computing, 2017, pp. 578–585.
- [138] F. Hock, P. Kortiš, 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 ICETA, 2016, pp. 91–96.
- [139] R.R. Gómez, Y. Nadji, M. Antonakakis, Towards designing effective visualizations for dns-based network threat analysis, in: Proceedings of the IEEE Symposium on Visualization for Cyber Security, 2017, pp. 1–8.
- [140] C. Jackson, A. Barth, A. Bortz, W. Shao, D. Boneh, Protecting browsers from DNS rebinding attacks, *ACM Trans. Web TWEB* 3 (1) (2009) 2:1–2:26.
- [141] K. Schomp, M. Allman, M. Rabinovich, DNS resolvers considered harmful, in: Proceedings of the 13th ACM Workshop on Hot Topics in Networks HotNets, 2014, pp. 16:1–16:7.
- [142] N.L.M. van Adrichem, N. Blenn, A.R. Lua, X. Wang, M. Wasif, F. Fatturrahman, F.A. Kuipers, A measurement study of DNSSEC misconfigurations, *Secur. Inform.* 4 (1) (2015) 8.
- [143] K.R. Babu, V. Padmanabhan, BDI based performance enhancement in DNSSEC, in: Proceedings of the 14th International Conference on Distributed Computing and Internet Technology ICDIT, 2018, pp. 121–126.
- [144] J. Bau, J.C. Mitchell, A security evaluation of DNSSEC with NSEC3, in: Proceedings of the 17th Annual Network and Distributed System Security Symposium (NDSS), San Diego, CA, 2010.
- [145] T. Chung, R. van Rijswijk-Deij, D.R. Choffnes, D. Levin, B.M. Maggs, A. Mislove, C. Wilson, Understanding the role of registrars in DNSSEC deployment, in: Proceedings of the 2017 Internet Measurement Conference (IMC), London, United Kingdom, 2017, pp. 369–383.
- [146] M. Jalalzai, W. Shahid, M. Iqbal, Dns security challenges and best practices to deploy secure dns with digital signatures, in: Proceedings of the 12th International Bhurban Conference on Applied Sciences and Technology (IBCAST), 2015, pp. 280–285.
- [147] F. Weimer, Florian weimer's home page [cited 12.02.2018], <http://www.enyo.de/fw/>.
- [148] J.M. Spring, C.L. Huth, The impact of passive dns collection on end-user privacy, Securing and Trusting Internet Names.
- [149] Farsight security DNSDB [cited 12.02.2018], <https://www.farsightsecurity.com/tools/dnsdb-scout/#faq>.
- [150] . ISC, Internet systems consortium [cited 12.02.2018], <https://www.isc.org/>.
- [151] M. Jonker, A. Sperotto, R. van Rijswijk-Deij, R. Sadre, A. Pras, Measuring the adoption of ddos protection services, in: Proceedings of the 2016 Internet Measurement Conference, 2016, pp. 279–285.
- [152] B.H. DNS, Black hole dns list [cited 12.02.2018], <http://www.malwaredomains.com/bhdns.html/>.
- [153] T.D.-B. project, Dns-bh – malware domain blocklist [cited 12.02.2018], <http://www.malwaredomains.com/>.
- [154] Z. Tracker, Zeus domain blocklist [cited 12.02.2018], <https://zeustracker.abuse.ch/>.
- [155] . Alexa, Alexa top sites [cited 12.02.2018], <https://aws.amazon.com/alexa-top-sites/>.
- [156] Common crawl [cited 12.02.2018], <https://commoncrawl.org/>.
- [157] R.R.G. rgomez30@gatech.edu, Active dns project [cited 12.02.2018], <https://activednsproject.org/>.
- [158] O. van der Toorn, R. van Rijswijk-Deij, B. Geesink, A. Sperotto, Melting the snow: Using active DNS measurements to detect snowshoe spam domains, in: Proceedings of the IEEE/IFIP Network Operations and Management Symposium, NOMS, 2018, pp. 1–9.
- [159] Y. Li, R. Xiao, J. Feng, L. Zhao, A semi-supervised learning approach for detection of phishing webpages, *Optik-Inter. J. Light Electron Opt.* 124 (23) (2013) 6027–6033.
- [160] R. Jenke, A. Peer, M. Buss, Feature extraction and selection for emotion recognition from eeg, *IEEE Trans. Affect. Comput.* 5 (3) (2014) 327–339.
- [161] A. Khormali, J. Addeh, A novel approach for recognition of control chart patterns: Type-2 fuzzy clustering optimized support vector machine, *ISA Trans.* 63 (2016) 256–264.
- [162] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, E. Duchesnay, Scikit-learn: Machine learning in python, *J. Mach. Learn. Res.* 12 (2011) 2825–2830.
- [163] J.D. Rodriguez, A. Perez, J.A. Lozano, Sensitivity analysis of k-fold cross validation in prediction error estimation, *IEEE Trans. Pattern Anal. Mach. Intell.* 32 (3) (2009) 569–575.
- [164] N.C. Fofack, S. Alouf, Modeling modern dns. caches, Modeling modern dns caches, in: Proceedings of the 7th International Conference on Performance Evaluation Methodologies and Tools, ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2013, pp. 184–193.
- [165] H. Shulman, M. Waidner, One key to sign them all considered vulnerable: Evaluation of DNSSEC in the internet, in: Proceedings of the 14th USENIX Symposium on Networked Systems Design and Implementation (NSDI), Boston, MA, 2017, pp. 131–144.
- [166] W. Lian, E. Rescorla, H. Shacham, S. Savage, Measuring the practical impact of DNSSEC deployment, in: Proceedings of the 22th USENIX Security Symposium (Security), Washington, DC, 2013, pp. 573–588.
- [167] C.T. Decchio, J. Sedayao, K. Kant, P. Mohapatra, Quantifying and improving DNSSEC availability, in: Proceedings of 20th International Conference on Computer Communications and Networks, ICCCN, 2011, pp. 1–7.
- [168] S. Hao, H. Wang, A. Stavrou, E. Smirni, On the DNS deployment of modern web services, in: Proceedings of the 23rd IEEE International Conference on Network Protocols ICNP, 2015, pp. 100–110.
- [169] X. Chen, H. Wang, S. Ren, X. Zhang, Maintaining strong cache consistency for the domain name system, *IEEE Trans. Knowl. Data Eng.* 19 (8) (2007) 1057–1071.
- [170] H. Shang, C.E. Wills, Piggybacking related domain names to improve DNS performance, *computer, Comput. Netw.* 50 (11) (2006) 1733–1748.
- [171] A.R. Kang, S.H. Jeong, S.Y. Ko, K. Ren, A. Mohaisen, Transparency in the new gTld era: Evaluating the DNS centralized zone data service, in: Proceedings of the Fourth IEEE Workshop on Hot Topics in Web Systems and Technologies, HotWeb, 2016, pp. 54–59.
- [172] J. Liang, J. Jiang, H. Duan, K. Li, J. Wu, Measuring query latency of top level DNS servers, in: Proceedings of the 14th International Conference on Passive and Active Measurement PAM, Hong Kong, 2013, pp. 145–154.
- [173] A. Mohaisen, M. Bhuiyan, Y. Labrou, Name server switching: Anomaly signatures, usage, clustering, and prediction, in: Proceedings of the 15th International Workshop on Information Security Applications, WISA, Jeju Island, Korea, 2014, pp. 202–215.
- [174] B. Ager, W. Mühlbauer, G. Smaragdakis, S. Uhlig, Comparing DNS resolvers in the wild, in: Proceedings of the Internet Measurement Conference (IMC), Melbourne, Australia, 2010, pp. 15–21.
- [175] Google public dns | google developers [cited 12.02.2018], <https://developers.google.com/speed/public-dns/>.
- [176] Cloud delivered enterprise security by opendns [cited 12.02.2018], <https://www.opendns.com/>.
- [177] Dns survey: 2010 [cited 12.02.2018], <http://dns.measurement-factory.com/surveys/201010/>.
- [178] K. Schomp, T. Callahan, M. Rabinovich, M. Allman, On measuring the client-side DNS infrastructure, in: Proceedings of the Internet Measurement Conference (IMC), Barcelona, Spain, 2013, pp. 77–90.

- [179] J. Mauch, Open resolver project, in: Presentation, DNS-OARC Workshop, Dublin, May 12, 2013, p. 2013.
- [180] Open resolver project [cited 12.02.2018], <http://openresolverproject.org/breakdown.cgi>.
- [181] Shadowserver [cited 12.02.2018], <https://dnsscan.shadowserver.org/stats/>.
- [182] D. Wessels, A. Mohaisen, Open resolvers in com/net resolution, in: Domain Name System Operations, Analysis, and Research Center, DNS-OARC, Tech. rep., 2014.
- [183] Y. Takano, R. Ando, T. Takahashi, S. Uda, T. Inoue, A measurement study of open resolvers and dns server version, in: Internet Conference (IEICE), 2013.
- [184] Z. Durumeric, E. Wustrow, J.A. Halderman, Zmap: Fast internet-wide scanning and its security applications, in: Proceedings of the 22th USENIX Security Symposium (Security), Washington, DC, 2013, pp. 605–620.
- [185] J. Park, A. Khormali, M. Mohaisen, A. Mohaisen, Where are you taking me? behavioral analysis of open dns resolvers, in: 2019 49th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN), IEEE, 2019, pp. 493–504.
- [186] L. Hendriks, R. de Oliveira Schmidt, R. van Rijswijk-Deij, A. Pras, On the potential of ipv6 open resolvers for ddos attacks, in: Proceedings of the 18th International Conference on Passive and Active Measurement PAM, 2017, pp. 17–29.
- [187] L. Pan, X. Yuchi, Y. Chen, Mitigating ddos attacks towards top level domain name service, in: Proceedings of the 18th Asia-Pacific Network Operations and Management Symposium APNOMS, 2016, pp. 1–4.
- [188] Z. Qin, C. Xiao, Q. Wang, Y. Jin, A. Kuzmanovic, A cdn-based domain name system, *Comput. Commun.* 45 (2014) 11–20.
- [189] B. Greschbach, T. Pulls, L.M. Roberts, P. Winter, N. 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, 2017.
- [190] K. Schomp, M. Rabinovich, M. Allman, Towards a model of DNS client behavior, in: Proceedings of the 25th USENIX Security Symposium (Security), Austin, TX, 2016, pp. 263–275.



Aminollah Khormali is a Ph.D. student of Computer Engineering at the University of Central Florida. He has obtained his M.Sc. in CS from UCF 2019, M.Sc. in Mechatronics Engineering from the K. N. Toosi University of Technology, in Tehran, Iran, in 2015. At UCF, he is working in the area of computer and network systems security. In particular, he explores the science of detecting and analyzing large-scale vulnerabilities in the context of network systems, IoT, cryptocurrency, and deep learning.



Jeman Park obtained his Ph.D. from the Department of Computer Science at the University of Central Florida in 2020, where he has been a member of the Security and Analytics Lab (SEAL). He obtained his B.Sc. degree in Computer and Communication Engineering from Korea University, in Seoul, South Korea, in 2016. His work has been focused on privacy, DDoS detection and mitigation, DNS security, anonymous network (Tor), and IoT security.



Hisham Alasmary is an Assistant Professor at King Khalid University. He obtained his Ph.D. from the Department of Computer Science at the University of Central Florida, and his M.Sc. degree in Computer Science from The George Washington University, in Washington, D.C., USA, in 2016. At UCF, he was a member of the Security and Analytics Lab (SEAL), where his work has been in the area of computer and network systems security. In particular, he explores the science and practice of threat information sharing and IoT security.



Afsah Anwar is a Ph.D. student at the Department of Computer Science at the University of Central Florida. He has obtained his Bachelors of Technology degree in Electronics and Communication Engineering from the Jamia Millia Islamia (JMI), in New Delhi, India, in 2014. At UCF, he is a member of the Security and Analytics Lab (SEAL), where he is working in the area of computer systems security. He builds automation tools for analyzing and mitigating vulnerabilities and their lifecycles in the context of large-scale software systems, mobile systems, and IoT.



Muhammad Saad is a Ph.D. student in the Department of Computer Science at the University of Central Florida. At UCF, Saad is a member of the Security Analytics and Research Lab (SEAL) advised by Prof. David Mohaisen. His research is on the security of distributed systems with an emphasis on the blockchain attack surface. His work has appeared in reputable venues including the International Conference on Distributed Computing Systems (ICDCS 2019) and Communication Surveys and Tutorial (CS&T 2020). He has won two best paper awards at Distributed Ledger of Things (DloT 2018) and Systems Journal 2019.



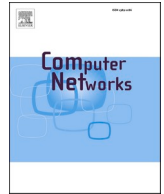
David Mohaisen is an Associate Professor at the Department of Computer Science at the University of Central Florida. He obtained his Ph.D. from the University of Minnesota in 2012, was an Assistant Professor at SUNY Buffalo, a Senior Scientist at Verisign Labs, and a Research Engineer at ETRI. At UCF, he directs the Security and Analytics Lab (SEAL), where he works in the broad area of computer security and privacy with applications to malware detection, distributed denial of service, critical infrastructure, Internet of Things, and blockchain systems. He was awarded the Summer Faculty Fellowship from the US AFOSR (2016) and several best paper awards (IEEE ICDCS 2017, WISA 2014, and CNS 2013). He is an Associate Editor of IEEE Transactions on Mobile Computing (TMC) and has been on the organizing committee of various top conferences. He is a Senior Member of ACM (2018) and IEEE (2015).

Update

Computer Networks

Volume 188, Issue , 7 April 2021, Page

DOI: <https://doi.org/10.1016/j.comnet.2021.107814>



Corrigendum to “Domain name system security and privacy: A contemporary survey” Computer Networks Volume 185 (2020) 107699

Aminollah Khormali, Jeman Park, Hisham Alasmmary¹, Afsah Anwar, Muhammad Saad, David Mohaisen^{*}

University of Central Florida, United States of America

The authors would like to apologize for any inconvenience caused.

DOI of original article: <https://doi.org/10.1016/j.comnet.2020.107699>.

^{*} Corresponding Author.

E-mail address: mohaisen@cs.ucf.edu (D. Mohaisen).

¹ The authors regret to inform that Hisham Alasmmary has done part of this work while affiliated with King Khalid University. Affiliation modification for Hisham Alasmmary: Part of this work was done while the authors was an Assistant Professor at King Khalid University, Saudi Arabia.

<https://doi.org/10.1016/j.comnet.2021.107814>