A Survey on Malicious Domains Detection through DNS Data Analysis

YURY ZHAUNIAROVICH, Qatar Computing Research Institute, HBKU, Qatar
ISSA KHALIL, Qatar Computing Research Institute, HBKU, Qatar
TING YU, Qatar Computing Research Institute, HBKU, Qatar
MARC DACIER, Eurecom, France

Malicious domains are one of the major resources required for adversaries to run attacks over the Internet. Due to the important role of the Domain Name System (DNS), extensive research has been conducted to identify malicious domains based on their unique behavior reflected in different phases of the life cycle of DNS queries and responses. Existing approaches differ significantly in terms of intuitions, data analysis methods as well as evaluation methodologies. This warrants a thorough systematization of the approaches and a careful review of the advantages and limitations of every group.

In this paper, we perform such an analysis. In order to achieve this goal, we present the necessary background knowledge on DNS and malicious activities leveraging DNS. We describe a general framework of malicious domain detection techniques using DNS data. Applying this framework, we categorize existing approaches using several orthogonal viewpoints, namely (1) sources of DNS data and their enrichment, (2) data analysis methods, and (3) evaluation strategies and metrics. In each aspect, we discuss the important challenges that the research community should address in order to fully realize the power of DNS data analysis to fight against attacks leveraging malicious domains.

CCS Concepts: • **Security and privacy** → **Network security**; Malware and its mitigation; • **Networks** → *Naming and addressing*; • **Computing methodologies** → *Machine learning approaches*;

Additional Key Words and Phrases: Malicious domains detection, Domain Name System

ACM Reference Format:
Yury Zhauniarovich, Issa Khalil, Ting Yu, and Marc Dacier. 2018. A Survey on Malicious Domains Detection through DNS Data Analysis. *ACM Comput. Surv.* 1, 1, Article 1 (February 2018), 35 pages.
https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

It is well known that the Internet is being used continuously to run attacks against different targets. Benign services and protocols are being misused for various malicious activities: to disseminate malware, to facilitate command and control (C&C) communications, to send spam messages, to host scam and phishing webpages. Clearly, it is very important to detect the origins of such malvolent activities, be it by identifying an URL, a domain name or an IP address. Many approaches have been proposed for such purpose: analysis of network traffic [144, 168], inspection of web content [38, 54], URL scrutiny [106], or using a combination of those techniques [76, 105]. On top of these, one of the most promising directions relies on the analysis of the *Domain Name System* data.

0360-0300/2018/2-ART1 $15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

Domain Name System (DNS) protocol is an essential part of the Internet. It maps tough-to-remember Internet Protocol (IP) addresses to easy memorable domain names. Detection of malicious domains through the analysis of DNS data has a number of benefits compared to other approaches. First, DNS data constitutes only a small fraction of the overall network traffic, what makes it suitable for analysis even in large scale networks which cover large areas. Moreover, caching, being an integral part of the protocol, naturally facilitates further decrease the amount of data to be analyzed, allowing researchers to analyze even the DNS traffic coming to Top Level Domains [29]. Second, the DNS traffic contains a significant amount of meaningful features to identify domain names associated to malicious activities. Third, many of these features can further be enriched with associated information, such as AS number, domain owner, etc. providing an even richer space exploitable for detection. The large amount of features and the vast quantity of traffic data available have made DNS traffic a prime candidate for experimentation with various machine learning techniques applied to the context of security. Forth, although the solutions to encrypt DNS data, like DNSCrypt [4] exist, still a large fraction of DNS traffic remains unencrypted, making it available for the inspection in various Internet vantage points. Last but not least, sometimes researchers are able to reveal attacks at their early stages or even before they happen due to some traces left in the DNS data.

The purpose of this paper is to survey all the approaches that aim at detecting domains involved in malicious activities through the analysis of DNS data. To do so, we have built a comprehensive bibliography by collecting papers from several sources. First, we have crawled 4 major digital libraries, namely, ACM[1], IEEEXplore[2], Springer[3] and Scopus[4], feeding them with a search string consisting of keywords relevant to the area. Second, we asked credible experts to provide us the most pertinent articles. Third, we extracted from these papers the references which were not included so far in our compiled list. Additionally, we continued to monitor major conferences for any relevant new work appearing in the area. Note that the focus of this paper is not limited to domains involved in specific types of malicious activities, as done in [26, 50, 56, 84], that provide surveys specifically about botnets; or in [86, 175], [147] and [137], that cover areas of phishing, web spam and malicious URLs detection correspondingly.

We have carefully read each paper in our study list and extracted the information that could help us to cover the targeted research topic. The first observation we have made, is that this research area is relatively new. The seminal paper [161], which led to the area as we know it today, dates back to 2005. Authored by Florian Weimer, it was the very first published paper not only to consider using DNS records to detect malicious domains but also to propose a practical solution to obtain large amounts of data amenable to various types of analysis. In order to position the numerous pieces of work that have followed, we propose a general framework (represented in Figure 1) to describe the various components required to implement a DNS based detection technique. It involves the following key components.

DNS Data Collection. DNS data could be collected at different locations of the DNS architecture as well as be available with different granularity. For example, they could be gathered at the recursive DNS server of a company or Internet Service Provider (ISP), or at higher-level authority servers. They may be available in the form of detailed DNS query/response logs, or only in aggregated forms. The location and granularity of the data can reveal different behaviors related to malicious domains, and thus have a significant impact on the intuition and design of the detection algorithms.

[1] http://dl.acm.org
[2] http://ieeexplore.ieee.org
[3] https://rd.springer.com/
[4] https://www.scopus.com/

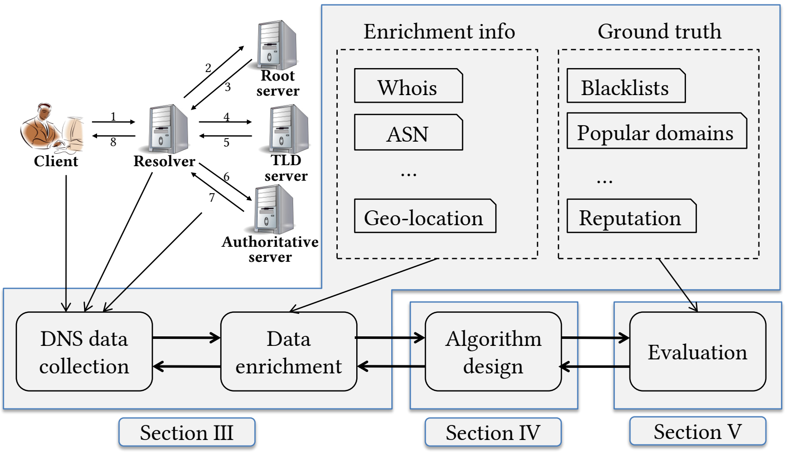

Fig. 1. The general process to design a DNS data-based technique to detect malicious domains.

Data Enrichment. To get a more comprehensive view of malicious activities, DNS data often needs to be enriched by integrating networking and application data from various sources. Typical data sources used for this purpose include domain registration records, autonomous system numbers and geo-location information of IPs hosting domains.

Algorithm Design. A detection algorithm identifies a set of potentially malicious domains based on DNS data, enrichment information and, possibly, intelligence on existing known benign and malicious domains. Existing machine learning algorithms (supervised, semi-supervised and non-supervised) are often adapted in this context, relying on various intuitions about the behavior of malicious domains.

Ground Truth. A ground truth of malicious and benign domains is needed both in the algorithm design phase and in the evaluation phase. Supervised and semi-supervised detection algorithms rely on known malicious and benign domains to train a machine learning model and tune important parameters. The evaluation of detection algorithms is also greatly influenced by how the ground truth set is collected, cleaned and applied.

Evaluation Methodology. Malicious domain detection imposes unique challenges that are not observed in typical machine learning problems, including its highly dynamic nature and the adaptiveness of attackers. Therefore, besides following standard evaluation methodologies from the machine learning community, additional evaluation criteria and methods need to be adopted to reflect the true effectiveness of a malicious domain detection scheme in practice. For example, it is highly desirable to evaluate the robustness of the approaches against adaptive attackers who could change their behaviors deliberately to evade detection.

A DNS-based malicious domain detection technique can be characterized by the above five key components. Reading the relevant articles, we paid particular attention to the information about (i) the sources of DNS data, enrichment data, and ground truth data, (ii) the extracted features and how they are used in various approaches, (iii) the evaluation metrics and strategies. The collected information constitutes the core of Sections 3, 4, and 5. Reading the papers and using our domain expertise, we identified a number of problems and challenges in the area. In each of these sections, we also discuss the challenges or unsolved problems faced by the research community. In Section 2 we provide necessary background on DNS and malicious activities leveraging DNS, while Section 6 concludes our discussion.

2 DOMAIN NAME SYSTEM BACKGROUND

This section aims at setting a common background on DNS and providing necessary shortcuts. Readers who are familiar with the topic can probably skip this section and move on to Section 3.

2.1 Domain Name System Operation

The *Domain Name System* is a hierarchical decentralized naming system that decouples the physical location (i.e., IP address) of a service and its logical address (i.e., its domain name), so that one can connect to the service using only its domain name. The DNS protocol has been introduced in November 1983 (IETF RFCs 882 [114] and 883 [115], later superseded by RFCs 1034 [116] and 1035 [117] correspondingly), and now it is an essential part of the Internet infrastructure as exemplified by the attack in October 2016 against the DynDNS provider [97]. By overloading their DNS servers with spurious requests, the attack prevented an extremely large portion of users from connecting to the Internet resources they needed access to. In this section, we briefly describe the key DNS concepts; the interested reader is referred to the IETF RFC 1034 [116] for more details.

Domain names are organized as a suffix tree structure called domain namespace. The root of this tree is the domain called *root* represented with a zero length label. The dot character is used in domain names to divide hierarchy levels. Parts between the dots are called labels. It should be noted that the trailing dot separating root domain is usually omitted; in the rest of the paper we follow this convention. The farthest right label is named the *Top Level Domain* (TLD), such as “com”. The domain directly on the left of a TLD is a *Second Level Domain* (2LD), e.g., “example.com”. *Fully qualified domain name* (FQDN), e.g., “www.example.com”, identifies a single node within the domain tree and is associated with the resource information composed of separate Resource Records (*RR*s). A Resource Record is defined by an **owner** (a domain name where the RR is found), a **type** field (an encoded 16 bit value that specifies the type of the RR), a **time-to-live** (TTL) value (time in seconds during which an RR should be cached), and an **RDATA** field, whose content and semantics depends on the value of the **type** field. In this work we are interested in the following types of RR (see [117] for the whole list): `A`/`AAAA` stores an IPv4/IPv6 host address; `NS` points to the authoritative server storing the information about the domain; `MX` is used to determine where a mail should be sent; `PTR` record maps a host’s IP address to its domain name or host name.

The domain namespace information (in the form of *resource records*) is stored in the hierarchical distributed database. Given the hierarchical structure, it is possible to divide it into separate *zones* (all domains under a particular node) and delegate the control under them to different authorities, which maintain this information in *zone files*. The Internet Corporation for Assigned Names and Numbers (ICANN) [17], a non-profit organization, is responsible for the creation of TLDs and delegation of their control to companies called *registries*, who are in charge for all the domains ending with that particular TLD. Registries work in close collaboration with *registrars*, companies like GoDaddy, which sell second level domains to domain owners (*registrants*) and provide billing and customer support.

To query information from a DNS database, a client specifies in the request a domain name and what type of the resource record it wants to obtain. The main algorithm specifying how standard queries are processed, is described in the RFC 1034 [116]. For the sake of this paper, it suffices to say (see also Figure 1) that clients in need of a domain name resolution, e.g., in need of knowing which IP corresponds to a given domain name, use the service of a *resolver*. The resolver will run what is called a *recursive* query on behalf of clients contacting it. This means that it will do its best to eventually return the needed response to the client and, to do so, may send a number of queries to various name servers without the client being involved. Once the resolver obtains an IP for a given domain name, it will cache the information, and, in most cases, will not query the same information for other clients. The important point to note here is that, due to caching, the resolver is the only one to have a complete view of how many clients use its services to resolve a given domain name. If a resolver does not have an answer in its cache for a given request, it will look for the *authoritative name server* in charge of that domain name. This search usually starts by asking the so-called *root name servers.* Authoritative name servers typically do not respond to recursive queries but, instead, to *iterative* ones, providing information at their disposal and leave it up to the requester to continue his quest by following the lead provided.

2.2 Domain Name System Security

DNS security has received a lot of attention from the research community over the years. There are plenty of attacks, where DNS is involved, and an even bigger number of methods to detect them. In this section, we briefly outline the area of *DNS Security* splitting it into the four subareas entitled hereafter as follows: *Securing DNS*, *Securing Data Provided by DNS*, *Securing Users from Attacks Leveraging DNS Disingenuously*, *Securing Users from Attacks Leveraging DNS Genuinely.*

Securing DNS. Being a cornerstone technology of the Internet, all DNS components has been widely attacked and exploited by adversaries. The DNS infrastructure has been targeted by a number of denial of service attempts, the latest major case being the already mentioned attack on the DynDNS infrastructure [97]. The DNS software has been the subject of attacks for many years now. According to Hoglund and McGraw [72], one of the very first reported Linux worms, the ADM worm, was spreading in a stealthy way in 1999 thanks to a buffer overflow vulnerability in DNS servers. Better software security engineering techniques, large amount of replicas for key DNS servers, deployment of anti-DDoS mitigation tools are among the various solutions that have, quite successfully, been brought forward to secure DNS from these attacks.

Securing Data Provided by DNS. Attackers always try to subvert the data provided by legit DNS servers because this allows them to redirect traffic to controlled resources. In June 2008, two of the world's most important Internet regulatory web sites, ICANN and IANA, were hijacked [92], which led to the creation of a set of best practices [141] that registrars should implement in order to keep the domain names of their customers secure. DNS hijacking and DNS poisoning attacks had been known for more than 25 years, with the seminal work by Steven Bellovin, produced in 1990 but withhold from publication until 1995 [33] followed by the 2002 birthday paradox attack [22]. However, only in 2008 with the so-called Kaminsky's attack [23], people really started paying attention to them. A number of approaches have been proposed to detect such attacks but the ultimate protection comes with the ever wider deployment of DNSSEC. Attackers have also misused the popularity of some web sites by (re-)registering their domain names just after their expiration dates, usually taking advantage of the oversight of their legitimate owners who failed to renew the registration in due time [39]. In this case, attackers exploit what is usually known as the "residual trust" of these stolen domains [98, 102], collecting money from ads showed to regular customers of the domain, hijacking emails, or pushing malicious content to the fooled client machines [98, 102, 104]. Another group of attacks falling into this category is generally known as cybersquatting [164], when an attacker registers an Internet domain name somehow similar to a victim's domain name. Typosquatting is one of such attacks that exploits common mistakes made by the users when they type domain names in an address bar. Being very typical, this attack has received a lot of attention from the research community [24, 32, 83, 152]. Other attacks of this group include bitsquatting [124], soundsquatting [123], and combosquatting [87].

Securing Users from Attacks Leveraging DNS Disingenuously. A third class of DNS security threat has to do with the disingenuous use of the protocol. DNS is among the very few protocols allowed in, probably, every computing network. Not surprisingly, a number of malware samples and botnets have misused it to enable the communications between compromised hosts and their command and control servers. Various approaches have been exploited, e.g., by using some well known fields such as the free form TXT field, or by encoding commands in queried domain names [52]. The same techniques have been also used for data exfiltration and malicious payload distribution [81]. Last but not least, in a more recent past, attackers have taken advantage of the fact that DNS replies are UDP based and much larger than the queries sent. They leverage these features to mount denial of service attacks, dubbed Reflective Denial of Service attacks [136], in which genuine DNS servers are being used to flood victims with large amounts of unwanted replies.

Securing Users from Attacks Leveraging DNS Genuinely. In this survey we concentrate on the attacks that leverage DNS genuinely to make them more resilient using the properties and features of the DNS protocol. To run malicious campaigns, mischievers need various kinds of services hosted in remote servers. In the early days, it was a common practice for malware to hardcode the IP addresses of the servers to receive orders or to exfiltrate data. That practice was abandoned very rapidly because the capture of a single malware sample could lead to the extraction of all these IPs, what was enough to shut down the whole botnet. It became clear that these servers needed to be able to move across the IP space. This is exactly what DNS had been made for. Moreover, in order not to be blacklisted, domain names should also have to move across the domain name space. There are two main techniques that are used to achieve this agile behavior: **Domain-Flux** and **IP-Flux** (or **Fast-Flux**). The former refers to the strategy having several FQDNs associated with one IP address. Using a *Domain Generation Algorithm* (DGA), a malware is able to dynamically generate new domain names (see [146] for a taxonomy on DGAs), usually as a function of the date and time. This technique makes it difficult, short of having reverse engineered the DGA, to block the domain names used by a given botnet since these domains have a very short lifetime. The latter (IP-Flux) is characterized by the continuous change of IP addresses associated with a particular domain name. In this case, a malware builds a *Fast Flux Service Network* (FFSN) [73] consisting of hundreds or even thousands of IP addresses assigned to a given domain name. When such a domain is queried, it is resolved to these IPs, which are frequently changed, thus, protecting the real location of the malicious service. Usually, the large pool of rotating IP addresses are not the final destination of the request for the content, they are just stopovers, so to speak, to reach the final destination, possibly after several other stops. *Double-flux* networks are a more complex technique providing an additional layer of redundancy. Specifically, both the DNS A record sets and the authoritative NS records for a malicious domain are continually changed in a round robin manner and advertised into the fast flux service network. Clearly, these techniques can also be used in combination providing many-to-many relationship between FQDNs and IP addresses.

Although these techniques are aligned with the specification of the DNS protocol, malware have abused them in various ways to improve the mobility of their servers and, thus, their resilience. The good news is that these techniques leave traces within DNS data. Such traces give researchers important clues to develop detection approaches taking into account the changes in domain-IP mappings using the unique viewpoint provided by the observation of the DNS traffic. In this survey, we focus on the approaches that are designed to detect domains involved in such malicious activities through the analysis of the relevant traces left in DNS data.

3 DATA SOURCES DEFINITIONS

In this section we categorize different types of DNS data, auxiliary information and ground truth that are used in the schemes proposed in the literature. The way these data are collected has a significant impact on the underlying assumptions and intuitions of malicious domain detection

Table 1. Summary of the “Data Sources Definitions” Section

Component	Dimension	Sources and Related Works
DNS Data	*1. Where the Data is Collected* a) Host-Resolver b) DNS-DNS	**1a**: BDS [130], Choi et al. [47], BotGAD [45, 46], Lee and Lee [100], Krishnan et al. [93], Manadhata et al. [109], Yadav et al. [165, 166], Smash [171], Segugio [133, 134], Perdisci et al. [128], Oprea et al. [126], Stalmans and Irwin [148] **1b**: Exposure [35, 36], Notos [28], Khalil et al. [82], Kopis [29], Huang and Greve [77], Yu et al. [169], Gao et al. [63], Mishsky et al. [112]
	2. How the Data is Collected a) Active b) Passive	**2a** (*Sources: Thales [91]*): Holz et al. [73], Fluxor [127], Nazario and Holz [121], BDS [130], Konte et al. [89], Ma et al. [105], Felegyhazi et al. [57], Hao et al. [68], Predator [69], DomainProfiler [44] **2b** (*Sources: Farsight database [55]*): Choi et al. [47], BotGAD [45, 46], Manadhata et al. [109], Exposure [35, 36], Notos [28], Khalil et al. [82], Kopis [29], Huang and Greve [77], Yu et al. [169], Gao et al. [63], Yadav et al. [165, 166], Smash [171], Segugio [133, 134], FluxBuster [129], Perdisci et al. [128], Oprea et al. [126], Mishsky et al. [112]
Data Enrichment	*1. Type of the Enrichment Data* a) Geo-location b) ASN c) Registration records d) IP/domain black-/whitelists e) Associated resource records f) Network data	**1a** (*Sources: MaxmindDB [111]*): Seifert et al. [142], Exposure [35, 36], BotGAD [45, 46], Gao et al. [62, 63], Zou et al. [174] **1b** (*Sources: MaxmindDB [111], Team Cymru [16]*): Khalil et al. [82], DomainProfiler [44], Fukuda and Heidemann [61], Stevanovic et al. [150], Kopis [29] **1c** (*Sources: WHOIS [49], commercial [5, 6, 20]*): Felegyhazi et al. [57], Predator [69], Fluxor [127] **1d** (*Sources: see ground truth*): Notos [28], Prieto et al. [130] **1e** (*Sources: Thales [91], Farsight DNS database [55]*): Hao et al. [68], Prieto et al. [130] **1f** (*Sources: Censys [53], Shodan [14], Team Cymru [16]*): Nadji et al. [120], Prieto et al. [130]
Ground Truth	*1. Type of the Ground Truth* a) Malicious b) Benign	**1a**: Spamhaus [15], Yahoo Webspam Database [21], PhishTank [125], VirusTotal [159], McAfee SiteAdvisor [10], Malware Domains [153], Malware Domains List [9], UrlVoid [18], Wepawet [59], McAfee SiteAdvisor [10], Google Safe Browsing [8], Web Of Trust [19], Anubis [1] **1b**: Alexa top ranked domains [25], McAfee SiteAdvisor [10], Google Safe Browsing [8], Web Of Trust [19]

schemes. Table 1 presents a short summary of this section and itemizes relevant articles. Notice that the table is not exhaustive, it only includes the most relevant examples of sources and articles.

3.1 Sources of DNS Data

The collection of DNS data can be categorized along the following two orthogonal dimensions: (1) *where* and (2) *how* the data is collected.

Where the Data is Collected. Due to the distributed nature of the DNS infrastructure, multiple locations can be considered to collect information about DNS queries and replies. Among all servers involved, the resolver (as defined in Section 2) is unique as it is the only location which has access to queries coming directly from client machines. Therefore, in the following, we distinguish two specific cases for the sources of the data. We call the first one “**Host-Resolver**”. It refers to DNS data obtained by observing the communications between an end host and its resolver. The second is called “**DNS-DNS**” and refers to the data that can be obtained by observing the communications between two DNS servers (and one of them could, possibly, be a resolver).

How the Data is Collected. Obtaining information about existing associations between IPs and domain names at a given point of time can be done in two ways. One way is to resolve **actively** and regularly a large collection of domain names to obtain that information. Another way is to observe **passively** all the requests sent to DNS servers extracting the necessary data. In the following, we will distinguish these two methods as **Active vs. Passive DNS data collection.**

3.1.1 Where the Data is Collected.

Host-Resolver (Flows 1 and 8 in Figure 1). One major advantage of the data captured at the internal interface of a resolver is that it provides detailed information about the clients in terms of DNS queries and responses, which may directly link to certain types of malicious behaviors [45, 47, 109, 126, 130, 133, 134]. For example, hosts controlled by a botnet often have similar DNS query model in terms of both queried domains and temporal patterns. Choi et al. [45, 46] use the information "what host queries what domain" to build a matrix for every domain that shows what machine at what period of time has queried this particular domain. Such a representation is very handy because it allows the analysts to prune matrix both column-wise and row-wise to correct errors that could arise due to deactivation of a botnet part or if time-window parameter is misconfigured. In the Segugio system [133, 134], this information is used to build a host-domain graph representing "who-queries-what" relation between hosts and domains. It would be harder to observe such behavior patterns from DNS-DNS data due to caching by the intermediate servers. Another advantage of this source of data is the ease of access. Any company or research institute could directly deploy sensors at its own resolver(s) requiring no co-operation with other parties. Due to these reasons, many existing schemes for malicious domain detection are built on data from resolvers, in particular those whose features are tied to the behavior of individual hosts. It should be also mentioned that the approaches that use Host-Resolver DNS data, may be also adapted to detect malicious hosts, roughly speaking those ones which query malicious domains.

One limitation of sensors deployed at the internal interface of a resolver is that they can only see the behavior of hosts inside a single organization, which may not be comprehensive enough to establish patterns related to malicious activities. One notable exception is when the client chooses to use, as a resolver, a publicly available DNS server willing to serve recursive queries, such as Google Public DNS [7], OpenDNS [12], or Norton ConnectSafe [11]. Due to the sheer volume and diversity of hosts they interact with, the data collected at these resolvers is suitable to comprehensively reveal suspicious behaviors related to different kinds of attacks. The DNS resolvers of large ISPs also serve a large amount of individual users. They can be used for the same purpose. Unfortunately, DNS data logs from public DNS servers or ISP DNS servers are not easily accessible to the research community, often because of privacy concerns [71, 88, 172].

DNS-DNS (Flows 2 to 7 in Figure 1). On the other hand, queries observed by sensors deployed near other DNS servers usually see queries issued from several organizations. In the literature, the most frequent locations considered to observe DNS-DNS traffic are (i) at the authoritative name servers [29] including the servers responsible for TLDs [29, 154], and (ii) at the external interface of the resolvers [28, 35, 36, 82]. The closer the sensor to the roots of the DNS tree, the larger the visibility. The data collected from TLD servers could offer unique insights and early detection of newly emerged malicious domains. Note, that such logs would only reveal the existence of the requests but not the answers to them (i.e., the IPs requested) since TLD servers typically serve only iterative queries. Such signals would be hard to capture only from logs of resolvers. Getting logs from an authoritative server solves this issue but, due to caching, not all queries will be visible to that server. Therefore, the view provided by the logs of DNS servers higher in the DNS tree can quickly become rather coarse grained. The extreme case is the requests observed at the root servers which give almost a full visibility of all names queried over the Internet but none of the responses. Volumetric analysis of these requests is also heavily impacted by caching happening in the intermediate servers between the end clients and the root servers [62, 63]. Therefore, the features offered by the data captured at servers different from the resolvers are often limited. Furthermore, the logs from such domain servers cannot be easily obtained by researchers.

3.1.2 How the Data is Collected.

Active DNS data collection. To actively obtain DNS data, a data collector would deliberately send DNS queries and record the corresponding DNS responses [44, 73, 89, 91, 105, 121, 127]. The list of queried domains is built thanks to multiple sources, typical ones include popular domains lists such as the Alexa Top Sites [25], domains appearing in various blacklists, or those from the zone files of authoritative servers. Clearly, as the queries are issued by the data collector, they do not reflect the behavior of actual users. Instead, active DNS data mainly capture the DNS records of domains, e.g., the resolved IPs, canonical names, TTL of a record, etc. The major advantages of actively crawled DNS data are the flexibility and ease of use of the data collection method. Data collectors can easily control which domains to query. Additionally, active DNS can reveal abuse signals about domains before their actual malicious use. For example, active DNS collector can discover in zone files a potentially malicious domain that has been newly registered but not yet used [57, 69], while passive sensors cannot see it yet. Moreover, active DNS data are not linked to the behavior of individual users, and therefore, can be shared with the research community without any privacy concern. Meanwhile, due to the same reason, active DNS data could not be used to detect malicious domains with techniques that rely on user-level features (e.g., temporal statistics of user queries). If the DNS queries are issued only from a limited set of hosts, the collected data could be biased, and this is another limitation. Specifically, a domain could be associated to multiple IPs depending on the geo-location of the query issuer. Therefore, active DNS data may contain a limited small set of IPs that are a function of where the queries are issued.

Passive DNS data collection. Collecting DNS data passively is done by deploying sensors in front of DNS servers or by having access to DNS server logs to obtain real DNS queries and responses [28, 29, 35, 36, 45, 82, 109, 126, 133, 134]. Therefore, DNS data collected passively are more representative and more "revealing" in sense of a rich set of features and statistics that could be derived to identify malicious activities. For instance, the Kopis system [29] in order to build a requester profile and to assess requester diversity, requires information regarding every resolver that queried data about a particular domain from an authoritative or TLD DNS server. Further, if sensors are deployed in DNS servers of diverse organizations from different locations, DNS data collected passively are likely to be more comprehensive than the ones collected actively. This assumption is indirectly confirmed by Rahbarinia et al. [133, 134]. Their system performed better if training and testing was executed on the data from the same ISP than if obtained from different ISPs. Moreover, such approaches do not require an initial precompiled list of domains. On the other hand, sharing of such data could be hindered due to privacy concerns, especially if sensors are deployed between clients and resolvers. Therefore, the existing publicly available passive DNS datasets are collected after resolvers and usually provide only aggregated views of queries to hide individual activities. For example, the Farsight passive DNS database [55] does not contain the IP addresses of requesters. Furthermore, for a given domain and one of its resolved IPs, it offers only the timestamps of the first and last seen resolution and the total number of them in between. This is a trade-off between privacy protection, the ability of sharing and the utility of the data. Similar to the actively obtained DNS data, it would be impossible to build fine-grained user-level features from this dataset. However, we note that, as the aggregation is done over queries and responses due to the actual host/user activities, some important aggregated user statistics still could be derived that may be very useful for malicious domain detection. For example, it is still possible to observe a sudden increase of queries over a set of domains globally in a short period of time, even after aggregation. Such statistics would not be available from actively collected DNS data.

3.1.3 Challenges.

The challenges of access to DNS data faced by the research community lie in two aspects. The first is in the data collection phase. Though DNS traffic is present in all networks, collection of a datasets is not an easy task. As discussed earlier, it is relatively easy to set up DNS traffic sensors in a single organization's network (e.g., a campus network), but then the collected data could offer only a limited local view of global threats. The peculiarity of many existing DNS-based malicious domain detection techniques is that they work best in big data scenarios. Thus, they may not be able to produce meaningful results on datasets collected in small networks. Meanwhile, integrating data from DNS servers belonging to different organizations would often face significant bureaucratic/legal obstacles, due to the sensitive nature of DNS logs. The same is true if researchers would like to gain access to the data from public DNS servers or from ISPs.

Even a bigger challenge lies in data sharing. Unfortunately, security related data are notoriously sensitive and hard to share. Even if a researcher is able to gain access to DNS logs from an ISP, it would be extremely difficult to make the same data available to peers for validation. At the same time, scientific advances rely on validation of and comparison with the existing approaches. There were some attempts to compare new approaches with the previous ones (e.g., Rahbarinia et al. [133] compared their approach with Notos [28]), but current research significantly lacks extensive and systematic experimental validation and comparison of different techniques. The primary reason lies in the difficulty to make publicly available a set of common or comparable reference datasets. Although currently there are several publicly available DNS datasets, which have been collected passively (e.g., from Farsight [55]) or actively (e.g., Thales [91]), they cannot be used in many approaches, especially in those relying on client-side patterns [109, 126]. It should be also noted that despite some approaches may work on data collected both actively and passively (for instance, the one proposed by Khalil et al. [82] which relies on domain co-location information obtainable from both datasets), such a comparison has never been performed before.

Moreover, the researchers must ensure that the results obtained with a particular dataset can be generalized to all other possible datasets. Clearly, some datasets may have space or time peculiarities that can influence the results considerably. For instance, Yadav et al. [165, 166] grounded their approach on the insight that the domain names generated automatically have abnormal distribution of character frequencies and that the algorithmically produced names are usually unpronounceable for an english speaker. Although in general this may be true for the majority of domain names, there are many countries in the world, e.g., China or Russia, where such intuitions may not hold true. It may be required to adjust the model to the peculiarities of the region.

3.2 Sources of Data Enrichment

DNS data represents an important source of intelligence that has been successfully used by many approaches to discover and predict malicious activities. However, to provide deeper insights about malicious activities and to enhance the accuracy and coverage, the majority of the detection approaches presented in this survey utilizes external sources of data to enrich DNS information. For example, mapping the IP address to a hosting country enables some approaches to use the trustworthiness of the country as a feature in classifying the maliciousness of domains/IPs [150]. Generally, the sources of data enrichment can be classified by the **Type of Information** they provide.

3.2.1 Enrichment Information Types.

Geo-location. The geo-locations of IPs and domains are commonly used to understand the diversity of the origins of the DNS queries as well as of machines hosting the domains. Such kind of enrichment is seen in a large number of papers, e.g., in [35, 36, 112, 127, 150]. The most common source of IP geolocation information observed in the literature is the Maxmind database [111].

Autonomous system number (ASN). This source of information enables to understand the distribution and utilization of adversary resources [36, 108, 112]. For example, legitimate domains (except those using CDNs) are usually hosted on one or few ASNs as opposed to malicious domains which hop from one ASN to another to evade detection. ASN is a valuable source of information allowing to distinguish different types of Internet services (e.g., IPs only used by dedicated organizations vs. those belonging to cloud service providers). The information on the IP-ASN mapping can be found in the Maxmind database [111] or using the Team Cymru service [16].

Registration records. Even though domain registration records often are not verified by authorities, the information located there sometimes can be used as supportive evidence to link malicious domains controlled by the same adversary. Further, temporal information of registration records (e.g., their creation/expiration time) is critical to identify domains registered automatically in bulks to be used later for malicious activities. In fact, some previous works rely purely on registration records to identify malicious domains [57, 69, 70]. The registration records information is usually obtained from servers which provide access to it through the WHOIS protocol [49]. It should be mentioned, there is no common standard on the format of the data provided. Hence, researchers must develop custom parsers in order to extract the necessary data.

IP/domain blacklists/whitelists. Domains are also often checked against well-known IP/domain blacklists (more information about blacklists/whitelists will be given in Section 3.3). For example, Notos [28] checks how many of the IPs associated to a domain are blacklisted, which is expected to be an indicator of the maliciousness of this domain. Other approaches check if the related IPs/domains are blacklisted. For instance, Prieto et al. [130] considers a domain suspicious if its authoritative name server is blacklisted.

Associated resource records. It is possible to gain more information about a given domain or IP by exploring other RRs related to it that can be retrieved from the DNS database. For instance, Hao et al. [68] have shown that the distribution of DNS MX records in the IP space for malicious domains is different than that of the benign ones. Moreover, Prieto et al. [130] observed that domain names associated with a botnet usually do not have any associated MX record.

Network data. The IP/domain data can be also enriched with information from network activities [130], e.g., if a website is associated with a domain, what is the HTTP response, what ports are opened, etc. Researchers usually obtain such kind of information by developing their own probes or using the information provided by Internet-wide scanners such as Censys [53] or Shodan [14].

3.2.2 Challenges.

It is important to understand that the information associated with an IP or a domain does vary over time. For instance, the Maxmind database [111], which is used to enrich data with the geolocation and ASN information, is frequently updated. Therefore, the values of the features calculated using these data also change. This results in a number of challenges. First, since researchers often work with historical DNS data, they must rely on the enrichment information available at the same time frame when the DNS data was collected. For instance, if they calculate the number of countries that host a particular domain at a given date, they have to use the information from the Maxmind database available at exactly the same date. As an alternative, they can use the most recent available enrichment data. Any of the approaches may be valid, and researchers must clearly identify which is used. The second challenge is tightly connected to the first one. Given a large number of IP addresses, the fast-growing number of domain names and the frequent change of the corresponding enrichment data, the maintenance and management of the related historical information requires a lot of resources that may not be available for researchers.

3.3 Sources of Ground Truth

Practically, every approach to detect malicious domains requires high-quality ground truth for training and validation. The ground truth data in this area is associated with domains and can be divided according to the **Type**.

3.3.1 Type of Ground Truth.

Malicious Ground Truth. To get a ground truth of malicious domains, the dominant practice in existing works is to extract it from various public blacklists. Some of the blacklists are only about specific malicious activities, e.g., spams (Spamhaus [15], Yahoo Webspam Database [21]), phishing (PhishTank [125], OpenPhish [13]), while some others are more general and include domains/IPs involved in any kind of malicious activities, e.g., VirusTotal [159], McAfee SiteAdvisor [10], Malware Domains [153] and Malware Domains List [9]. Some of these sources, such as WoT [19], can also blacklist domains that are not, per se, associated with malicious activities. This is the case when the content of such web sites is considered inappropriate with respect to the policies in place for the specific blacklist considered (e.g., pornographic content, violence, racism, copyrighted material, etc.). Another source to build ground truth is proprietary blacklist/whitelists, or proprietary reputation systems deployed by anti-virus security companies (e.g., Symantec), whose availability to the general research community is quite limited.

Benign Ground Truth. Ground truth of benign domains in the literature is largely drawn from highly ranked popular domains. For example, Alexa top ranked domains [25] are commonly used[5]. Another common practice, at least when building an initial candidate set of benign domains, is based on the top level domains. For example, domains from “gov” and “mil” zones or those belonging to Google and Microsoft (used, e.g., in [174]), are generally considered more trustworthy than those from “com” or “info”. Additionally, some public cyber intelligence tools like McAfee SiteAdvisor [10], Google Safe Browsing [8] or Web Of Trust [19] report not only malicious and suspicious domains, but also benign ones and hence, can be also used to extract benign ground truth.

3.3.2 Challenges.

Malicious Ground Truth Challenges. Even though reputable blacklists generally provide robust evidences about blacklisted domains, they still have a number of subtle issues. First, a malicious domain can be malignant in different ways: spam, phishing, C&C, unethical, adult content, etc. Thus, the mere definition of the term “malicious” differs from one ground truth dataset to another. The ground truth collected for one approach may not work for another one that focuses on detecting domains involved in other types of malicious activities. Second, blacklists employ different collection methods. For instance, they may rely on crowd sourced data (e.g., PhishTank [125], Web of Trust [19]), may crawl and analyze website content (e.g., Wepawet [59]), may run malicious software in sandboxes and analyze accessed domains (e.g., Anubis [1]), may reverse botnet protocol and generate feed of names produced by DGAs (e.g, Conficker [99]), may be obtained using internal tools (e.g., Google Safe Browsing [8]) or may aggregate data from different sources (for instance,

[5] We will explain later the need to apply a supplementary filter to the Alexa lists because they do contain malicious domains as well.

UrlVoid [18] or VirusTotal [159]). Third, none of the blacklists is completely reliable. Sinha et al. [145] and Ramachandran et al. [135] showed that blacklists exhibit high false positives and false negatives rates. Some approaches address this by cross-checking domains in multiple blacklists. For example, Kheir et al. [85] built a ground truth dataset by voting on 3 different blacklists.

Benign Ground Truth Challenges. Although blacklists may contain false positives, generally a domain can be considered as malicious if it has appeared in a reputable blacklist. At the same time, building a bening domain ground truth is a far harder task. A domain cannot be deemed as benign simply because it is not present in any known blacklist. The large number of Internet domains (according to Verisign [156], in 2016 there were around 314 million 2LDs) makes it impossible to scan and check them regularly. Although this number is large, it represents only a very small portion of the total number of FQDNs in the Internet. Even worse, that number keeps growing every day. Therefore, a malicious domain may not be blacklisted because it did not expose malicious content when it was scanned or it has never been scanned.

Although the usage of top K Alexa domains [25] as benign ground truth makes sense (the administrators of popular web pages devote more effort to protect their resources), it is both limited and suffers from high false positive rate. The list contains only 2LD domains and does not provide any information about sub-domains, which makes it rather limited. Domains are ranked according to their popularity but not based on their security or safety, which leads to high false positive rate. It contains proxies to malicious web pages or even domains hosting malicious content. For instance, a quite popular 2LD, unblocksit.es (ranked 11550 as of April 1, 2016), offers to proxy access to other, potentially blacklisted, domains. This 2LD is not, per se, malicious since it can be used by legit users to try to circumvent censorship measures that they are facing. Similarly, malicious users can abuse this service as a safe haven to defeat known blocking mechanisms. Moreover, some malicious domains could appear among top K Alexa domains due to a burst of requests from a high number of infected clients querying them. Stevanovic et al. [150] cross-checked domains from Alexa top K domains with UrlVoid [18], the service which aggregates information from different blacklists. The results show that a relatively high percentage of domains (around 15% out of 10,000 top domains) is reported to be malicious by at least one blacklist.

Such impurities of benign ground truth negatively affect the accuracy of domain detection approaches. For instance, consider a malicious domain d that is mislabeled as benign in the ground truth, as it is in Alexa top K domains. A correct detection of d would be counted as a false positive incorrectly, causing the measured false positive rate higher than what it really is. At the same time, a malicious domain with a strong association with d may be missed due to the lack of associations with malicious domains, that negatively affects the true positive rate. To mitigate the impact of Alexa top K impurities, some approaches filter the domains before adding them to benign ground truth. For example, Rahbarinia et al. [133] consider only domains that consistently appear in Alexa top 1 million sites for one year. Similarly, Bilge et al. [35, 36] consider only domains older than 1 year as benign. Some other approaches, e.g. [28, 82], remove dynamic DNS service domains, such as no-ip.com, before building a ground truth of benign domains. As one can see, there is no consensus on what could or should constitute the ground truth for benign domains.

Common Challenges. One of the common issues is to understand what domain level to use for ground truth compilation: 2LD, 3LD, or FQDN. Some ground truth sources contain domains of a specific level, e.g., top K Alexa domains [25] mostly consists of 2LD domains. This creates trouble for approaches that focus on the domain levels different from those found in the ground truth. The relations between domains at different levels are also unclear. Should we consider any subdomain of a malicious/benign domain as malicious/benign? Should we consider a domain as malicious/benign if the majority of its subdomains are malicious/benign? Unfortunately, there is no definite “Yes” or “No” answer to these questions. It may be reasonable, to a certain extent, to answer “Yes” to these questions for 2LDs that belong to private organizations like Google or Facebook. However, the subdomains of dynamic DNS services such as no-ip.com and 3322.org may be totally unrelated and hence, cannot be assumed benign even if the vast majority of their subdomains is benign.

Another common issue, which we have identified in the literature, is the limited quantitative discussion of training and testing sets comprising the ground truth data. It has been shown (see [42, 160]) that an imbalanced training dataset may have considerable influence on the learning of a classifier and thus, may influence some of the measured metrics.

4 DESIGN OF DETECTION ALGORITHMS

To systematically present the approaches that have been used by the various lines of work in the past, we have opted to look at them from three distinct viewpoints:

Features: What features are used?
Method: What technology is the detection method based on?
Outcome: What outcome is produced?

The following subsections address each of these viewpoints separately, while Table 2 provides an overview of the section by itemizing only the most relevant examples and related articles.

4.1 Features

Feature extraction (a.k.a., feature engineering) is a challenging task which has a big impact on the quality (accuracy and robustness) of the detection approaches. Well-crafted features contribute considerably to the success of an approach, and on the contrary, poor features may ruin even good detection algorithms. On the other hand, even though a feature may have good predictive power leading to a high detection accuracy, if it can be easily forged by an attacker, the robustness of the detection approaches relying on it will be low. Therefore, successful detection approaches must take into consideration a delicate balance of accuracy and robustness when selecting their features.

Very few approaches simply parse Resource Records from DNS traffic and use values from specific fields as they appear. Instead, a multitude of treatments can be applied to these raw values before consuming them for detection purposes (average, standard deviation, max, min, rate, outlier, etc.). Furthermore, an external data, outside the DNS environment, may be used to enrich the initial dataset. Some approaches require to transform the DNS data into a distinct data structure, such as a graph, before using it in their detection methods. For instance, this is the case in the approach proposed by Lee et al. [100, 101], where a graph representing the communication sequences of clients with domains is built. The authors call it the Domain Name Travel Graph (DNTG) and use it to identify clusters of related domains that need to be considered by their detection method. In the approach proposed by Oprea et al. [126], another type of graph is built representing the association between host IP addresses and queried domains, while in the Khalil et al. [82] approach a graph captures the movement of domains in bulks among different ASNs.

The number of individual treatment, enrichment and preprocessing techniques is very large and going through each and every one is out of the scope of this paper. In order to present the state of the art in a systematic way, we distinguish consumed features at a higher level of abstraction. Specifically, we consider the following three dimensions to differentiate features:

1. Internal vs. Contextual features
2. DNS dataset Dependent vs. Independent features
3. Mono vs. Multi domains features

4.1.1 Internal vs. Contextual Features.

The distinction between internal and contextual features is quite similar to the one proposed by

Table 2. Summary of the "Design of Detection Algorithms" Section

Component	**Dimension**	**Examples and Related Works**
Features	*1. Internal vs. Contextual* a) Internal b) Contextual	**1a** (*Examples: Domain average TTL value, Domain name label features, Association-based features*): Exposure [35, 36], Perdisci et al. [128], FluxBuster [129], Stalmans and Irwin [148], Notos [28], Pleiades [30], BotGAD [45, 46], Phoenix [140], Zou et al. [174], Oprea et al. [126], GMAD [101], Segugio [133, 134], Stevanovic et al. [149] **1b** (*Examples: Number of ASNs to which the IP addresses of a domain belong to, Number of ASNs, Historical association of domains to IP addresses*): Chiba et al. [44], Stalmans and Irwin [148], BotGAD [45, 46], Hu et al. [75], Khalil et al. [82], Kopis [29], Notos [28]
	2. DNS Dataset Dependent vs. DNS Dataset Independent a) Dependent b) Independent	**2a** (*Examples: Number of IP addresses assigned to a domain, Number of common ASNs shared by a pair of domains*): Hu et al. [75], Konte et al. [89], Perdisci et al. [128], FluxBuster [129], Khalil et al. [82] **2b** (*Examples: Hit-count of a domain, N-gram distributions of letters in a domain name*): Exposure [35, 36], Notos [28], Pleiades [30], Marchal et al. [110]
	3. Mono Domain vs. Multi Domains a) Mono b) Multi	**3a** (*Examples: Number of countries which host a domain, Number of distinct IP addresses, N-gram distribution of characters in a domain name, Average TTL value*): Stevanovic et al. [150], Exposure [35, 36], Kopis [29], Fukuda and Heidemann [61], DomainProfiler [44] **3b** (*Examples: Related historic domains features, Client sharing ratio between the connected domain names, Number of shared ASN*): Segugio [133, 134], Khalil et al. [82], Zou et al. [174], Smash [171], Thomas and Mohaisen [154], Notos [28], GMAD [101], Pleiades [30]
Detection Methods	*1. Type of Detection Methods* a) Knowledge based b) Machine learning based 1) Supervised 2) Semi-supervised 3) Unsupervised c) Hybrid approaches	**1a** (*Examples: Degree of co-occurrences between known malicious and unknown domains, Similar NXDomain behavior, Distribution of character frequencies in domain names*): Choi et al. [47], Krishnan et al. [93], Guerid et al. [65], Yadav et al. [165, 166], Holz et al. [73] **1b1** (*Examples: Naive Bayes, Decision Tree, Random forest, SVM, Neural networks*): FluXOR [127], Exposure [35, 36], DomainProfiler [44], Stalmans [148], Woodbridge et al. [163], Kopis [29], Fukuda and Heidemann [61], Hu et al. [75], Mentor [85] **1b2** (*Examples: Cluster-and-label, Belief propagation, Shortest path, Other graph-based approaches*): Huang and Greve [77], DNSRadar [107], Manadhata et al. [109], Mishsky et al. [112], Zou et al. [174], Khalil et al. [82], GMAD [100, 101], Lee and Lee [100], Gao et al. [62, 63], Felegyhazi et al. [57] **1b3** (*Examples: K-means, X-means, Hierarchical clustering, Agglomerative clustering, Fast unfolding*): Jiang et al. [79], Stevanovic et al. [150], BotGAD [45, 46], Thomas and Mohaisen [154], Smash [171], Berger and Gansterer [34] **1c** (*Examples: combination of machine learning approaches, mix of machine learning and knowledge based techniques*): Perdisci et al. [128], Oprea et al. [126], FluxBuster [129], Pleiades [30], Notos [28], Segugio [133, 134], Yu et al. [169]
Outcome	*1. Malicious Behavior Agnostic vs. Malicious Behavior Specific* a) Agnostic b) Specific	**1a**: Zou et al. [174], Manadhata et al. [109], Oprea et al. [126], GMAD [101], Khalil et al. [82], Mishsky et al. [112] **1b** (*Examples: DGA detection, FFSN detection*): Yadav et al. [165, 166], Haddadi et al. [66], Grill et al. [64], Pleiades [30], DeepDGA [27], Fu et al. [60]

Perdisci et al. [128] to divide features into *passive* and *active*. According to the authors, *passive* features are the ones "that can be directly extracted from the information collected by passively monitoring the DNS queries" from resolvers, while "*active* features need some additional external information to be computed". Since we do consider, elsewhere, the possibility to collect the data passively or actively, we felt this terminology could be misleading and therefore, we opt for the different terms, namely internal and contextual, which are described below.

Internal features. These features can be extracted from DNS Resource Records alone. No external complimentary data source is required. However, they may be and most of the time are transformed before being fed into the detection method. For instance, the "domain average TTL value" used in [35, 36, 128, 129, 148] is an example of this type of features. Additionally, features extracted from domain names, which are popular in DGA detection and attribution ([28, 30, 45, 73, 140, 155]), belong to this category. Moreover, association-based features popular in graph-based approaches [100, 101, 126, 133, 134, 149, 174], are usually built using internal DNS features.

Contextual features. On the other hand, *contextual* features are built from the combined DNS and external information sources. For instance, to calculate "the number of ASNs to which the IP addresses of a domain belong to" ([44, 45, 75, 148]) the information about the IP-AS mapping is required. In other example [82], the authors use similarity score calculated over the number of different AS numbers to assign a weight to domain-domain associations. Zhang et al. [171] also exploit associations inferred from WHOIS data for domain clustering.

We note that some contextual features require to query resources controlled by attackers. For instance, Prieto et al. [130] use domain web presence as one of the features, i.e., every time when a new domain appears in their list they check if a webpage is available for this domain. One more special type of contextual features employs the enrichment using DNS data itself. For instance, Prieto et al. [130] checks if a domain has an associated MX record. Hence, the usage of such type of features may warn an attacker that the domain is under scrutiny. However, it is not always required to interact actively with the domains. Such a data sometimes can be obtained from the systems like Thales [91], Censys [53] or Shodan [14].

Whereas the usage of internal features has a number of benefits, mostly in terms of simplicity, their ability to capture the information that has been shown to be significant to distinguish between good and bad domain names, is limited. For instance, the registration time for a given domain is often a very important feature but it cannot be obtained solely from the DNS data. It has been shown that sometimes attackers register domains in bulk several months before the start of malicious activities [69]. Detection of such registration patterns enables researchers to proactively detect malicious domains as done in [57, 69]. However, that information usually is not available for country code TLDs (ccTLD) because ccTLD registries very rarely offer access to their zone files. Therefore, the existence of a domain can remain unknown until it is queried for the very first time, and at this moment it may be possible (sometimes but not always) to retrieve that information by querying a WHOIS server. This makes the approaches relying on such features inapplicable for a very large amount of domains. Similarly, some other useful enrichment information can be hard to obtain due to limited accessibility, privacy concerns, excessive cost, etc. However, despite all these issues the usage of contextual information allows researchers to extract more meaningful features and hence, provide broader coverage of malicious behavior signals.

4.1.2 DNS Dataset Dependent vs. Independent features.

Based on our review of the literature, we believe it is important to distinguish between the features that are influenced by specific DNS datasets and those that are independent from the DNS dataset in hand. We call them DNS Dataset Dependent Features (DDD) and DNS Dataset Independent Features (DDI) respectively. The rationales behind these two classes are linked to the validation phase. The performance of an approach solely relying on DDD features is highly influenced by the chosen dataset. Thus, to evaluate the quality of such methods, it is very important to perform cross-dataset validation, using datasets from different places, for different periods, of different sizes, etc. (see Section 5.3 for more). On the contrary, approaches relying on DDI features are more stable and can be run equally in different environments.

DNS dataset dependent features. For instance, "the number of IP addresses observed as being assigned to a domain" during the observation period is a DDD feature, because its value depends on the specific dataset [75, 89, 128, 129]. Similarly, "the number of observed common ASNs shared by a pair of domains" feature used by Khalil et al. [82] to build an association between domain names, is also dataset dependent because a graph built using this association hinges on where and how a dataset has been collected.

DNS dataset independent features. On the other hand, the "hit-count of a particular domain in popular search engines" [35, 36] is a DNS dataset independent feature because it does not depend on what one can see in the DNS dataset chosen. Similarly, the "n-gram" distribution of a domain name [28, 30, 110] is DNS dataset independent since it does not hinge on the chosen dataset.

4.1.3 Mono vs. Multi Domains Features.

Mono domain features. Mono Domain features are extracted for every single domain. For example, "the number of countries which host a given domain" [28, 29, 44, 61, 150], is an example of a Mono Domain feature. One of the advantages of using this type of features is that the approaches rely on them can be trained and operate on completely different datasets.

Multi domains features. Domain association features calculated over a pair of domains, which are used in many graph-based and clustering approaches [82, 133, 134, 154, 171, 174], are examples of Multi domains features and so are the ones used, for instance, in [28, 30, 101]. We note that the approaches relying on *Multi Domains* features usually require bigger datasets to work properly. Indeed, an association between two arbitrary domains may be indirect, hence in order to build such an association intermediate domains should be also included into consideration in order for the approach to work properly.

4.2 Detection Methods

We have identified two main paradigms in the detection methods we are considering. In the first, the method may benefit from some external expertise to figure out how to discriminate between good and bad domains. This expertise is implemented by means of various heuristics and does not use machine learning techniques. Therefore, we call the approaches under this paradigm **Knowledge Based methods**. In the second case, whereas the authors may have some examples of malicious and benign domains at their disposal, they have no a priori understanding on how to distinguish between the two. They rely on data driven algorithms to help automatize the discrimination process, therefore, we call the approaches under this paradigm as **Machine Learning Based methods**.

In general, the approaches belonging to the former category appear earlier than the latter. In early research efforts, through the analysis of data, researchers identified characteristics that allowing to distinguish malicious domains from benign ones. However, with the lapse of time adversaries adapted their behavior causing the degradation of the approaches' detection abilities, what forced researchers to look for more descriptive characteristics. Such races resulted in the situation when the number of characteristics required to be considered in one model, became unmanageable, pushing researchers to look towards the **Machine Learning Based methods** able to automatically derive knowledge from high-dimensional data.

With a further development of the field, researchers started to employ stacking of methods. In order to produce a list of malicious domains, these methods involve several steps when the output of one method is passed as an input to the following one. So as these techniques employ different detection methods including machine learning and knowledge based, we call them as **Hybrid approaches**.

4.2.1 Knowledge Based Approaches.

To detect domains involved in malicious activities, knowledge based approaches rely on expert insights. Such insights can be obtained through measurement studies, which explore anomalies relevant to malicious domain activities. There is a number of such studies in the literature [37, 40, 47, 64, 65, 93, 118, 130, 139, 158, 170]. For instance, Sato et al. [139] observed malicious domains belonging to one malware family tend to be queried simultaneously. Hence, by measuring a degree of co-occurrences between known malicious and unknown domains and by comparing the result with some threshold, it is possible to detect new malicious domains. Hyunsang Choi exploited the same observation in his works [45–47]. Krishnan et al. [93] and Guerid et al. [65] observed the communities of bots in a network tend to exhibit similar patterns in terms of DNS queries that can not be resolved by the DNS infrastructure.

Unfortunately, this family of approaches have limitations. Experts can intentionally or most often unintentionally be biased. For instance, Grill et al. [64] built their approach on the observation that the DGA malware makes a lot of DNS resolutions in order to find the right domain to communicate with. Therefore, for hosts infected with this type of malware the amount of DNS resolutions is larger than the amount of subsequent communications. Comparing the ratio between them with a manually set threshold allowed the authors to detect hosts infected the malware. However, modern browsers try to predict users' Internet behavior and resolve ahead of time some domains, even if they are never queried. Hence, in such a scenario, if the threshold is not adjusted automatically, the approach will generate false positives since such behavior was unknown to the experts at the time of analysis. Furthermore, experts usually are not good at analyzing high-dimensional data because for a human being it is difficult to grasp all the correlations and dependencies between features extracted from the data.

4.2.2 Machine Learning Based Approaches.

The majority of the methods developed to detect malicious domains are data-driven with machine learning algorithms at their core [150]. Generally, machine learning algorithms allow computers to learn on data without being explicitly programmed [113, 138]. Depending on what data is used for learning, existing machine learning techniques can be generally divided into three subcategories:

- Supervised learning
- Semi-supervised learning
- Unsupervised learning

Supervised learning algorithms. These algorithms require the complete training set to be labeled, i.e., every feature vector corresponding to a sample of data must be associated with a label representing a class this sample belongs to. With respect to the topic of our paper, this means every domain name in the training set must be explicitly labeled as either malicious or benign. However, considering the amount of domains typically observed during the training period of experiments, it is almost impossible to label all of them correctly. Therefore, usually in case of supervised learning approaches the training data set is trimmed to contain only those labeled with high confidence. Interested readers may refer to [90] for a review of supervised learning algorithms. Supervised machine learning approaches such as [29, 35, 36, 44, 61, 75, 85, 122, 127, 148] are quite popular in this area due to their simplicity, automatic selection of the most relevant features and effectiveness. Indeed, researchers relying on such approaches only need to extract features from raw data and train a classifier on a labeled dataset. Application of the trained classifier to new data is straightforward. For example, DomainProfiler [44] uses 55 features extracted considering related IP addresses and domain names. The Random Forest algorithm is applied to discover abused domains. Antonakakis et al. [29] also employs Random Forest. However, in this work the features are extracted from the passive DNS data of authoritative name servers.

Unfortunately, supervised learning approaches have several drawbacks. First, they require a labeled dataset to train. It is not easy to obtain complete and fully correct dataset because of the fickle nature of DNS and blacklist data. As discussed in Section 3.3, manual labeling is time-consuming and does not result in extensive training datasets. Automatic labeling using information from different white- and blacklists likewise is prone to incorrect data inclusion [51, 94, 95, 135, 143, 145, 150]. Second, supervised learning approaches are more vulnerable to overfitting to a particular dataset. If the labeled dataset is biased, this may unintentionally cause a classifier to learn incorrect distributions of the feature variables. Moreover, in a real feed of DNS data only a portion of domains can be assigned with labels. In practice, the vast majority of samples are not labeled and thus, can not participate in the process of classifier learning making training dataset inconsistent.

Semi-supervised learning algorithms. The semi-supervised learning algorithms [41, 173] have been proposed to overcome such limitations. They learn both from labeled and unlabeled data. The unlabeled data helps a machine learning algorithm to modify or reprioritize hypothesis obtained from a labeled dataset [173]. Yet, the adoption of such algorithms often is quite challenging and requires more effort from researchers. We refer to [173] and [41] for more information about semi-supervised learning algorithms. Graph-based inference methods are among the most popular approaches under this category [77, 82, 100, 101, 109, 112, 126, 174]. For instance, Manadhata et al. [109] detected malicious domains applying the belief propagation algorithm to a host-domain graph extracted from the enterprise HTTP proxy logs[6]. Assuming that malicious hosts more likely communicate with malware domains, while benign hosts may only occasionally query malicious domains, and having a feed of initially malicious and benign domains, the authors, using the belief propagation approach were able to assess the marginal probability of unknown domains in the graph to be malicious. In [174], the authors predicted malicious hosts and domains applying their method on two types of graphs. The first, Domain Query Response Graph (DQRG), is built using the information from DNS query-response pairs: clients' IP addresses are connected with the queried domain names which on their turn are associated with the returned domains' IP addresses. The second, Passive DNS Graph (PDG), is built using domain names, their canonical connections and corresponding IP addresses (CNAME and A resource records) extracted from passive DNS data. Then, belief propogation was applied on these graphs. Contrary to [109], where all benign domains' initial score has the same value, Zou et al. [174] assigned the value based on their rank in the Alexa top K list. Mishsky et al. [112] applied the Flow algorithm on a domain-IP graph. However, this graph includes, besides weighted domain-IP edges commonly used in this area, also domain-domain and IP-IP edges that represent "tell me who your friends are and I will tell you who you are" relation.

The Cluster-and-Label semi-supervised learning technique is also widely used [57, 62, 63, 100, 101] in the area. Gao et al. [62, 63] proposed an approach to detect malicious domains through clustering based on co-occurrence patterns. Clearly, queries to DNS system from the same malicious agents do frequently co-occur, e.g., when a bot tries to resolve algorithmically generated domain names in order to find the IP address of a master. In this case, the same domain names will frequently pop-up together in DNS resolver logs. The authors exploited this observation in the following way. At first, they performed coarse-grained clustering of the traffic. They selected a time window and for every anchor domain (malicious domain from a labeled dataset) measured how often it co-occurs with other domains within the selected time window. They calculated two metrics: terms frequency that shows how often other domain names are queried together with the anchor domain,

[6] We consider this work because the paper assures the same algorithm can be applied to DNS data.

and inverse document frequency showing how rare other domains are met across all the windows. Using the predefined thresholds for both metrics, the authors selected coarse-grained clusters associated with every anchor domain. Further, to perform fine-grained clustering, every domain is assigned with a bit vector whose length is equal to the number of times anchor domains are met during the observation period. A bit in this vector is set if the query to the domain happens within a small time window with the query to the anchor one. Later, these vectors are clustered using X-means to select fine-grained clusters. Jehyun Lee and Heejo Lee proposed a new approach to build a graph representing a sequence of client-domain communications, which they called *Domain Name Travel Graph (DNTG)* [101]. A node in this directed graph represents a domain, while an edge is added between two nodes if the corresponding domains have been queried sequentially by the same client. The weight of an edge grows with the increase of the number of transitions between those domains, while the direction of an edge shows the order of the transition. An edge is also associated with the client sharing ratio score that represents Jaccard similarity of the sets of the clients queried the domains. After the graph is built, it is clustered using the values assigned to the edges and some predefined thresholds. Then, the authors mark all the domains in the clusters containing blacklisted domains as malicious.

At the same time, this type of algorithms is not a silver bullet in the case of limited ground truth. The usage of unlabeled data does not always help, hence, researchers must put additional efforts in the validation of the proposed methods. Also, the problems related to obtaining a correctly labeled dataset are relevant here as well.

Unsupervised learning algorithms. The unsupervised learning methods [34, 45, 46, 79, 154, 171] have been introduced not only to eliminate the dependence on labeled datasets. Unsupervised learning approaches, aka clustering techniques [78], automatically divide domains into clusters using only the internal properties of data. In theory, by careful selection of the features which exibits a completely different behavior for malicious and benign domains, it is possible to enable clustering algorithms to divide the provided samples into two clusters. Then, a researcher decides what cluster contains malicious and benign domains [45, 46, 150]. However, some approaches, e.g., [154, 171], do not follow this path and make a step further. They group domains across several dimensions related to different malicious behaviors, and then select the clusters of malicious domains by correlating the identified groups among each other.

Although such approaches have a clear benefit in terms of independence over the labeled data, they are not very common in the literature. We believe this is mainly due to the fact that these techniques are the most difficult to design. Additionally, given that labeled datasets usually exist in this area (althogh neither complete nor fully correct), researchers prefer to explore supervised and semi-supervised methods which are easier to employ.

4.2.3 Hybrid Approaches.

Despite the fact that a single detection algorithm can be categorized according to the provided classification, the majority of the existing real-world approaches are hybrid and employ several algorithms of different types to produce a result. This can be a combination of machine learning techniques [28, 30, 126, 128, 129]. For instance, such approach is used in the Notos system [28]. It trains 5 meta-classifiers during the first stage to evaluate the closeness of a domain to the predefined group of domains (Popular, Common, Akamai, CDN and Dynamic DNS) using a supervised learning technique. Then the calculated closeness scores are used as features for the second-stage supervised learning algorithm. Oprea et al. [126] combines a semi-supervised method (belief propagation) with a supervised learning algorithm (linear regression). A mix of machine learning and knowledge based methods is also used in the area [133, 134, 169]. For instance, the Segugio system [133, 134] combines graph-based prefiltering with supervised machine learning. It works in the following way. At first, the system, using the DNS data collected before a recursive DNS resolver, builds a host-domain graph. Given a set of benign and malicious domains, and some heuristics, it performs filtering of this graph. It marks the known domain nodes as benign and malicious respectively leaving the rest as unknown. Similarly, the system labels host nodes as malicious if they query one of the malicious domains, and benign that resolve only the benign domains. All other machines are marked as unknown. After this, the system performs pruning of the graph removing: 1) machines querying 5 domains or less; 2) proxy hosts (machines quering substantially more domains than other machines); 3) domains that are queried by only one machine; 4) very popular domains (domains queried by a very large number of machines). Then, every domain node left in the graph is also assigned with the following properties: 1) a set of IP addresses the domain is pointed to during the observation window; 2) how long ago the domain was first queried with respect to the observation time window. Using this information Segugio calculates several features: 1) Machine Behavior Features (the fraction of known infected machines, the fraction of unknown machines, total number of machines); 2) Domain Activity Features (number of days a domain was actively queried during the last 2 weeks, the number of consecutive days a domain was queried); 3) IP Abuse Features (fraction of IPs associated to known malware domains during the selected time window, number of IPs and /24's used by unknown domains during time window). Using these features and supervised machine learning algorithms, the authors predict labels of unknowns.

4.3 Outcome

At the end, all we want to know is if a domain is malicious or not. However, the mere term *malicious* can be understood in different ways. For instance, some domains may be involved in spamming or phishing, serving C&C communications, or simply acting as proxies for other types of campaigns. Among many methods proposed, some are capable to recognize specific types of "maliciousness", whereas others are not able to explain why they adjudicate a certain domain is malicious or not. Therefore, in this paper we divide approaches according to the outcome of their operation between those detecting *specific malicious behavior* and those that are *agnostic to malicious behavior*.

Malicious behavior agnostic approaches. Roughly speaking, *malicious behavior agnostic approaches* do not try to capture particular malicious behavior. Instead, they base their intelligence on different type of *associations* between domains. The approaches of this type [82, 101, 109, 112, 126, 174] will predict maliciousness of domains exploiting connection with the domains constituting the ground truth. Such technique is called sometimes "guilty by association" [82]. If a domain has strong connection with a group of known malicious domains, then most probably, this domain is also involved in malicious activities. For instance, if adult-related domains are used as a ground truth, as a result such approaches will produce the list of domains of the same type, given that these domains make use of the same association. Similarly, if such approaches are fed with spam domains, they will predict domains related to spam activities. At the same time, only few blacklists report malicious domains of particular type, e.g., PhishTank [125] or Spamhaus [15]. Moreover, it is usual that the same infrastructure may be used for different malicious activities. Therefore, even if an approach is fed with a ground truth of particular type, the output may include other types of malicious domains. For example, an attacker may use a server with the same IP address that hosts different types of malicious domains. If an approach builds an association between domains according to the common IP addresses, it will establish a connection between these domains.

Malicious behavior specific approaches. On the contrary, *malicious behavior specific approaches* are built to capture specific features relevant to particular malicious behavior. For instance, there is a number of approaches that specifically try to capture lexical [66, 165, 166] or resolution [30, 64] features suitable for the detection of automatically generated domain names. Some of the approaches extract features detecting multiple malicious activities. So, Bilge et al. [35, 36] extract domain name based features which are relevant (although may be not perfect [27, 60]) for capturing DGAs, and DNS answer-based features (e.g., amount of different IP addresses, TTL values, etc.), which are apt for detection of domains exposing IP fluxing behavior.

4.4 Challenges

4.4.1 Feature Related Challenges.

Even though the process of finding meaningful features is not easy in other research areas as well, it is especially challenging in the field of malicious domain detection. Features are not only needed to be well crafted to separate benign from malicious domains, but also they have to be resilient to potential manipulation by miscreants. For example, certain DGAs produce easily recognizable names (e.g., “ccd2.cn”, “syx4.cn”, “oif1.cn”, etc.) and one could see this as a powerful feature to identify these malicious domain names. While this is currently true for a very limited number of DGAs, it is trivial for the attacker to render this feature inoperative by simply changing some parameters of the domain generation algorithm. On the other hand, a feature that takes into account the limited capacity of certain resources (e.g., number of public IP addresses) is more robust because it is harder to forge it without impacting negatively the attacker’s gain.

Unfortunately, it is not easy to evaluate the robustness of features in a systematic and measurable way. The importance of the problem has been recognized by many researchers, e.g., in [29, 44, 96, 101, 107, 122, 126, 129, 171]. However, up to our knowledge, none of the existing approaches provides a framework that can be used to evaluate quantitatively the robustness of features. Stinson et al. [151] presents a qualitative high level evaluation of the evadability of some botnet detection approaches. Others, such as Hao et al. [69], qualitatively discuss the robustness of some of the important features used in their approach. Nevertheless, providing a framework that offers qualitative and quantitative evaluation of the feature robustness remains an open problem that calls for attention from the research community. Such frameworks have to consider simultaneously the features forging complexity and their impact on attack utility. We argue such a framework could be an effective mean against adaptive attackers as it would help researchers and security experts to build detection tools leveraging features whose forging negatively impacts the attackers’ benefits.

4.4.2 Detection Methods Related Challenges.

Even though the effectiveness of a detection method is important and receives due attention in most of the approaches, its performance is somehow overlooked. However, deep performance analysis is as important as effectiveness analysis for practical consideration and real-world deployments. In real-world deployments, the amount and the rate of DNS traffic could be considerably larger than the datasets used in publications. Hence, detection approaches have to be scalable to work in such production systems. Moreover, some approaches require large datasets to train and to tune their detection algorithms. To address this problem, some authors propose to use distributed computing platforms such as Apache Hadoop [3] or Apache Giraph [2]. Others reduce their dataset sizes by filtering out data elements deemed to be less important. For example, Exposure [35, 36] filters out all domains from the Alexa Top 1000 domains [25] and those that have been queried less than 20 times during a predefined period of time. Unfortunately, such filtering may result in overlooking important sets of domains which could be potentially malicious. In such cases, we need a systematic performance evaluation that takes into account not only the complexity and scalability of a detection method but also the characteristics of the filtering preprocessing steps required for the needed data size reduction.

Next to the performance evaluation challenge, the second one faced by the malicious domain detection methods is related to the latency endured before the detection. Some approaches like [35, 36] rely on aggregated data or run in batch mode, and hence, they have to observe a number of DNS requests before being able to make a decision about the malicious status of a domain. However, the delay incurred by such approaches may render them ineffective against domains that serve malicious activities for short periods of time as is the case of domains fluxing. For example, Sheng et al. [143] showed that "63% of the phishing campaigns lasted for less than two hours". On the other hand, some approaches leverage real time features (as opposed to aggregates) and can flag domains on the fly. However, non-aggregated features are usually easier to forge comparing to aggregated ones. Both categories of approaches have advantages and limitations, and hence, the optimal selection of one over another is heavily influenced by the deployment environment.

The third challenge is linked with the adaptive nature of the adversaries. They continuously adapt their behavior to evade detection tools, and detection techniques have to regularly retrain and adapt their models to capture such changes. Moreover, this also means that the techniques themselves with the lapse of time become obsolete, making the corresponding approaches no longer possible to use.

The fourth challenge lies in the lack of any systematic way to quantitatively compare and contrast the effectiveness and the efficiency of various domain detection methods. To obtain reliable quantitative results, every approach should be reproducible and measurable. Reproducibility means the results can be regenerated given the same dataset used in initial training, while measurability means the use of quantitative metrics in evaluating effectiveness and performance. Unfortunately, the authors of approaches rarely share datasets and implementation code, possibly due to the privacy, proprietary, and sometimes security related issues, which makes it hard to reproduce the results and considerably complicates the comparison. One way to go over this challenge is to implement tools proposed in these works using information available in public sources such as papers and technical reports. However, the complexity of such tools is usually paramount and the public sources do not contain sufficient and detailed information to provide reasonable implementation of the approach.

4.4.3 Outcome Related Challenges.

As a result of an algorithm execution, the system predicts if a domain is malicious or not. However, a domain may be malicious in different aspects. For instance, in the obvious case a domain can be defined as malicious because it is used to send spams or to distribute malicious software. Unfortunately, what constitutes a malicious behaviour is not always that well defined. An example is domains hosting adult content. Some approaches, e.g., Predator [69], consider these domains as malicious because they are often used in spam-related campaigns. Others [95, 133, 165] consider such domains as benign. At the same time, it is shown that they are often a cause of higher false positive rates, especially if the ground truth contains this type of domains [43]. Generally, Wondracek et al. [162] confirmed adult domains are often used for malware distribution and aggressive marketing, and should not be blindly considered as benign. Hence, researchers should clearly identify in their works which domains are considered as malicious.

5 EVALUATION METHODS

As discussed in the previous section, the majority of DNS-based malicious domain detection approaches leverage machine learning concepts and techniques such as clustering and classification. Therefore, it is natural for them to use the evaluation metrics and strategies that have been developed and used by the machine learning community. However, this area has the unique challenge of adaptive attackers, who continuously change behavior to evade detection. This limits the time and the scope of the validation results and calls for adaptive evaluation strategies. In this section, we present the commonly used evaluation metrics and strategies, and articulate the unique challenges that researchers face when they validate malicious domain detection approaches. Table 3 provides a short summary of the information considered here.

Table 3. Summary of the "Evaluation Methods" Section

Dimension	Categories	Examples and Related Works
Metrics	*1. Metric Types* a) TPR/Recall b) FPR c) TNR d) FNR e) Precision f) Accuracy g) F1-score h) AUC	**1a**: Haddadi et al. [67], Segugio [133, 134], Krishnan et al. [93], DomainProfiler [44], Kopis [29], Manadhata et al. [109], Khalil et al. [82], Zou et al. [174], Yadav et al. [165, 166], Yadav and Reddy [167], Notos [28], Villamarin-Salomon et al. [158] **1b**: Haddadi et al. [67], Segugio [133, 134], Krishnan et al. [93], GMAD [101], Kopis [29], Manadhata et al. [109], Khalil et al. [82], Zou et al. [174], Yadav et al. [165, 166], Notos [28], Villamarin-Salomon et al. [158] **1c**: DomainProfiler [44] **1d**: Oprea et al [126], Stevanovic et al. [150], Qian et al. [132] **1e**: GMAD [101], DomainProfiler [44], Fukuda and Heidemann [61] **1f**: Hsu et al. [74], Stevanovic et al. [150], Fukuda and Heidemann [61] **1g**: DomainProfiler [44], Fukuda and Heidemann [61], Haddadi et al. [66] **1h**: DomainProfiler [44], Manadhata et al. [109], Pleiades [30], Huang and Greve [77], Exposure [35, 36], FluxBuster [129]
Evaluation Strategies	*1. Evaluation Types* a) Whole dataset b) One round train-test split c) Leave-p-out cross-validation d) K-fold cross-validation e) Cross-networks validation f) Cross-time validation g) Cross-blacklists validation	**1a**: Choi et al. [47], Villamarin-Salomon et al. [157], Felegyhazi et al. [57], Hu et al. [75], BotGAD [45, 46], Gao et al. [62, 63], ExecScent [122], Guerid et al. [65], Stevanovic et al. [150, 150], Smash [171], Predator [69], PsyBoG [96] **1b**: Lee and Lee [100], Haddadi et al. [66], Mentor [85], Fukuda et al. [61], Oprea et al. [126] **1c**: **1d**: Fluxor [127], Holz et al. [73], Notos [28], Sato et al. [139], Exposure [35, 36], Kopis [29], Pleiades [30], FluxBuster [129], Manadhata et al. [109], Zou et al. [174], DomainProfiler [44], Khalil et al. [82] **1e**: Segugio [133, 134], DNSRadar [107] **1f**: Segugio [133, 134], Yadav et al. [165, 166], Pleiades [30], DomainProfiler [44], Predator [69] **1g**: Stalmans and Irwin [148], Segugio [133, 134], Felegyhazi et al. [57]

5.1 Metrics

As mentioned earlier, evaluation is tightly coupled with the ground truth. For the purpose of this section, the ground truth consists of a set of domains labeled either as malicious or benign. Let P and N be the number of malicious and benign domains in the test set, respectively; TP (True Positives) and TN (True Negatives) be the number of correctly identified malicious and benign domains; and FP (False Positives) and FN (False Negatives) be the number of benign domains that have been incorrectly identified as malicious and the number of malicious domains that have been incorrectly identified as benign, respectively. The most commonly used evaluation metrics in this area are:

- **True Positive Rate (*TPR*) or Recall**: The ratio of the correctly identified malicious domains to the total number of malicious domains ($TPR = TP/P$); the higher the value is, the better ($TPR \in [0, 1]$).
- **False Positive Rate (*FPR*)**: The ratio of the benign domains flagged as malicious to the total number of benign domains ($FPR = FP/N$); the lower the value is, the better ($FPR \in [0, 1]$).
- **True Negative Rate (*TNR*)**: The ratio of the correctly identified benign domains to the total number of benign domains ($TNR = TN/N$); the higher the value is, the better ($TNR \in [0, 1]$).

- **False Negative Rate (*FNR*)**: The ratio of the malicious domains flagged as benign to the total number of malicious domains ($FNR = FN/P$); the lower the value is, the better ($FNR \in [0, 1]$).
- **Precision**: The ratio of the correctly identified malicious domains to the number of all identified malicious domains ($precision = TP/(TP + FP)$); the higher the value is, the better ($Precision \in [0, 1]$).
- **Accuracy (*Acc*)**: The ratio of the correctly identified domains to the whole size of the test set ($Acc = (TP + TN)/(P + N)$); the higher the value is, the better ($Acc \in [0, 1]$).
- **F1-measure or F1-score**: The harmonic mean of precision and recall ($F_1 = 2 * precision * recall/(precision + recall)$); the higher the value is, the better ($F_1 \in [0, 1]$).

During the design phase, a detection algorithm is tuned to identify the thresholds that optimize the desired metrics. However, some of these metrics are negatively correlated, i.e., enhancing the value of a desired metric may result in degrading the value of another one. For example, the desire to increase the *TPR* may result in the undesired increase of the *FPR*. Therefore, detection accuracy is usually assessed based on a discrimination threshold that reflects the dependency of *TPR* on *FPR*, which is called the **Receiver Operating Characteristics (ROC)** curve. The ROC graphical representation enables researchers to assess the achieved true positive rate once the value of false positive rate is fixed. Although a ROC curve is a good graphical representation, it cannot serve as a comparative quantitative metric. Therefore, the **Area Under the ROC Curve (AUC)** has been proposed as a quantitative comparison metric (e.g., [64]). In general, a system with higher AUC score is better. However, only few approaches report the AUC values what makes it difficult to compare and contrast different methods.

Finally, we note that some approaches use customized metrics to evaluate other important parameters of their system. For example, Khalil et al. [82] report the "expansion" as the number of newly detected domains for a given number of known malicious domains (seed). Hao et al. [69] defines "completeness" as the number of detected domains compared to other blacklists, and "delay" as the time it takes blacklists to identify a spammer domain after registration, while Ma et al. [107] use the time-lagging as a metric to evaluate how long it takes other public sources to blacklist a detected domain.

5.2 Evaluation Strategies

Malicious domain detection approaches use different evaluation strategies. Most of them borrow strategies from the machine learning community, where cross-validation is one of the most popular technique. In cross-validation, the dataset is split into training and testing parts and multiple rounds are performed using different partitions to reduce variability. The partitioning could be exhaustive as in the case of the **leave-p-out cross-validation** or non-exhaustive as in the **k-fold cross-validation**. In leave-p-out cross-validation, p out of the total n observations are used for testing and the remaining observations are used for training. The results are averaged over all possible p combinations out of the n observations, which makes it difficult to apply in practice due to the large number of rounds. Therefore, this strategy is almost not used in the area. The k-fold cross-validation is more practical and hence, more popular (e.g., [28–30, 35, 44, 73, 77, 82, 93, 109, 127, 129, 139, 174]). According to this strategy the ground truth dataset is divided into k equal parts, where $k - 1$ parts are used for training and the remaining part is used for testing. The experiment is repeated k times, changing every time the part used for testing, and the final score is obtained as an average of all the k rounds. Other popular strategies include: (i) **Validation against the whole dataset** (e.g., [43, 45, 47, 57, 62, 63, 65, 69, 75, 122, 150, 157, 171]). According to this strategy, all predictions are verified against the whole ground truth data. This method is popular in unsupervised approaches, where there is no need for a training set. (ii) **One round train-test split** divides ground truth into two non-overlapping training and testing sets (e.g., [61, 66, 67, 85, 100, 126]). For example, the authors in [66, 67] use 70% of the ground truth for training and the rest 30% for testing.

Although these validation strategies provide quite reliable results from the machine learning community's point of view, they have issues when applied to malicious domain detection. For instance, attackers and benign users in different parts of the world may have different behavior and hence, different organizations have different traffic profiles. For example, the traffic in a governmental organization network is different from that of a supplying company. Thus, the model produced from the data in one part of the world may not be suitable for the data produced in other parts of the world. Moreover, attackers usually change their behavior over time to avoid being detected, therefore, testing on time periods closer to the training time period may produce better results. Finally, an approach may be good at detecting domains belonging to one specific botnet, while performing poorly for other malware types. To address these issues, several cross-dataset strategies were proposed and applied in this area: (i) **Cross-networks validation**, in which the training and testing datasets are separated in space, i.e., training and testing datasets are collected at different locations (e.g., [107, 133]); (ii) **Cross-time validation**, in which training and testing datasets are collected at different time periods (e.g., [30, 44, 69, 133, 165, 166]); (iii) **Cross-blacklists**, in which training and testing datasets are collected from different malware blacklists (e.g., [133, 148]). Ideally, a system should be trained and tested on completely different data separated both in terms of time and space.

5.3 Challenges

The first challenge malicious domain detection approaches face with, lies in the difficulty of new knowledge validation. The majority of the approaches validates effectiveness only against part of the ground truth, the testing set, which is usually a small subset of the whole dataset. However, most of the approaches do not systematically show how to validate the predicted malicious domains that are not part of the ground truth. A few detection approaches have partially addressed this challenge (e.g., [30, 69, 82, 150, 171]) by one or a combination of the following strategies:

Cross-inspection. Newly detected malicious domains are checked against sources of intelligence other than those used for the ground truth collection. However, it is clear that no combination of blacklists covers all existing malicious domains, otherwise, the new approach would generate already known data and thus, would be redundant. Hence, if this technique is applied, and the approach identifies new malicious domains it is impossible to validate them.

Manual content inspection. The content of newly detected domains is manually checked for malicious traces. In addition to being not scalable [127], manual inspection is not reliable [131]. The cost of manually crawling and investigating the content of the potentially large number of newly detected domains is prohibitive. Therefore, only a small set of randomly selected domains is usually checked, while the content of the rest remains unverified.

Automatic content inspection. Newly detected domains are fed into tools that perform automatic content scanning. Automatic verification is not always reliable because the traces of automatic tools could be detected by malicious domain owners or the malicious domain could be proxied by look-like-benign domains [48, 80, 103]. Additionally, malware domains may simply not expose their malicious services to the public but rather target only specific visitors.

Cross-time validation. Newly detected domains are periodically checked after prediction against reputable commercial and public blacklists, or using manual content checking. However, one caveat of this strategy is that malicious owners may completely abandon domains which had been predicted to be malicious or simply have them behaving benignly [143]. Indeed, there

> is significant evidence that some attackers verify the presence of their resources in public blacklists before launching an attack [31]. Another caveat is that this strategy is affected by the dynamic maliciousness status of some domains over time. Domains being malicious at the detection time may become benign later and vice versa. For example, in February 2016, Linux Mint web server was hacked and used to distribute malicious content [119] but later it has been regained and cleaned. The first transition (malicious to benign) negatively impacts true positives, while the second transition (benign to malicious) does the same with true negatives.

The second challenge is the absence of a publicly available reference dataset. Although there was an attempt to provide such a dataset (see Los Alamos DNS Dataset for APT Infection Discovery Challenge [58]), this practice was not widespread. Having a publicly available reference test set is an important step towards providing a benchmark to compare the effectiveness of various approaches, and it can help researchers to further advance the area in a more systematic way. The absence of a reference dataset combined with difficulties in sharing code makes it hard to repeat experiments for systematic comparison of different approaches. However, we admit that attackers change behavior over time to avoid detection moving from one network to another and adjusting their attack methods. Therefore, it may be hard, if not impossible, to collect a reference dataset that covers different deployment environments and survives the dynamic behavior of adversaries. Complementary to this issue is the absence of a reference ground truth data. Different approaches use different sources. As discussed before, such sources may target different malicious activities and hence, cover different domains. Additionally, the lack of sharing among different sources could increase the gap. For example, in [43], out of the 296 and 192 malicious sites that SiteAdvisor and Safe Web have identified, only 8 are common. That is, evaluating based on a ground truth collected from one source may differ from that based on a ground truth collected from another.

The third challenging issue is in building a unified approach for metric calculation. A real DNS data usually consists of domains which are not all covered by white- and blacklists. This leaves the treatment of some metrics to the discretion of researchers. For instance, one can consider all domains appearing in blacklists as malicious while treating all others as benign. Other may take into the consideration only labeled part of domains out of the whole DNS dataset performing metrics computation. Such approaches may considerably influence the results. Along the same line, filtering of a dataset also influences the evaluation results. Indeed, filtering of domains applied in some approaches (e.g., [35, 36, 79, 128, 150]) may influence both positive and negative sides. For instance, in [35, 36] Bilge et al. filtered out domains "queried less than 20 times during the entire monitoring period" (because some aggregated statistics simply do not work if there are less than this amount of queries). However, among these domains there may be a number of malicious ones. Therefore, filtering out these domains will increase the amount of false negatives. Similarly, the detection rate is also impacted once long-lived domains are removed [35, 36].

Last but not least, sometimes the approaches in this area are compared using the accuracy metric. This metric is not reliable in case of imbalanced datasets, i.e., those where the number of samples of one class is considerably higher than that of other. Such datasets are quite common in the area. Indeed, it is easy to find a large amount of benign domains, e.g., by using Alexa Top 1,000,000 domains [25], while the number of malicious domains is limited by the ones available in blacklists. Therefore, it is better either to use the metrics insensitive to imbalanced datasets (e.g., AUC or F1-measure) or to balance the sets before measuring accuracy [42, 160]. Finally, even though in the majority of works the results are reported using the TPR and FPR scores, these approaches can be barely compared because the TRP and FPR metrics depend on each other. Therefore, in order to compare two methods one of the metrics' values in both approaches should be fixed.

6 CONCLUSION

DNS data carry rich traces of the Internet activities, and are a powerful resource to fight against malicious domains that are a key platform to a variety of attacks. In this paper, we presented a large body of research efforts on utilizing DNS data to detect malicious domains. Table 4 summarizes our systematization scheme and findings. As our survey shows, to design a malicious domain detection scheme, one has to consider the following major questions: (1) data sources (Section 3): what types of DNS data, ground truth and auxiliary information are available; (2) features and data analysis techniques (Section 4): how to derive features to match intuitions of malicious behaviors, and what types of detection techniques the malicious domain discovery problem can be mapped to; (3) evaluation strategies and metrics (Section 5): how well standard evaluation methodologies fit the detection problem in a specific application context, whether there is a need for additional evaluation strategies that better capture the operational settings when a detection scheme is deployed in practice, how to evaluate the robustness of a technique given the adaptive nature of attackers, and what metrics to use for these purposes.

Our analysis identifies several significant challenges that hinder the advances of the field. First, in terms of data availability, we observe that large-scale real DNS data logs are seldom publicly available, and sharing of such information across organizational boundaries often faces legal, privacy-related or bureaucratic obstacles. Also, in terms of ground truth, there is no widely agreed practice in the community how to build ground truth from noisy public intelligence. Second, in terms of features and detection techniques, we have highlighted a number of challenges such as the resilience of the features, the adaptability of algorithms to evading attackers, and the interpretation of the results. Third, in terms of evaluation strategies and metrics, current research lacks established theoretical foundations and systematic empirical frameworks to evaluate the robustness of malicious domain detection schemes.

Providing a deep overview of the area, identifying existing challenges, and sharing our insights obtained doing the research in this field, we hope this survey will facilitate future research and development of methods and applications to fight against attacks leveraging malicious domains.

REFERENCES

[1] [n. d.]. Anubis. ([n. d.]). http://anubis.iseclab.org/ Service discontinued.

[2] [n. d.]. Apache Giraph. ([n. d.]). Retrieved 23-04-2017 from http://giraph.apache.org/

[3] [n. d.]. Apache Hadoop. ([n. d.]). Retrieved 23-04-2017 from http://hadoop.apache.org/

[4] [n. d.]. DNSCrypt - Official Project Home Page. ([n. d.]). Retrieved 05-11-2017 from https://dnscrypt.org/

[5] [n. d.]. DomainHistory.net: Detailed Domain Name Information and Archives in One Place. ([n. d.]). Retrieved 12-04-2017 from http://www.domainhistory.net/

[6] [n. d.]. DomainTools: Whois Information. ([n. d.]). Retrieved 12-04-2017 from http://whois.domaintools.com/

[7] [n. d.]. Google Public DNS. ([n. d.]). Retrieved 15-02-2017 from https://developers.google.com/speed/public-dns/

[8] [n. d.]. Google Safe Browsing. ([n. d.]). Retrieved 10-08-2016 from https://developers.google.com/safe-browsing/

[9] [n. d.]. Malware Domain List. ([n. d.]). Retrieved 20-08-2016 from https://www.malwaredomainlist.com/

[10] [n. d.]. McAfee SiteAdvisor. ([n. d.]). Retrieved 10-08-2016 from http://www.siteadvisor.com/

[11] [n. d.]. Norton ConnectSafe. ([n. d.]). Retrieved 15-02-2017 from https://dns.norton.com/

[12] [n. d.]. OpenDNS Premium DNS. ([n. d.]). Retrieved 15-02-2017 from https://signup.opendns.com/premiumdns/

[13] [n. d.]. OpenPhish. ([n. d.]). Retrieved 27-04-2017 from https://openphish.com/

[14] [n. d.]. Shodan: The Search Engine for ... ([n. d.]). Retrieved 03-04-2017 from https://www.shodan.io/

[15] [n. d.]. Spamhaus. ([n. d.]). Retrieved 25-08-2016 from https://www.spamhaus.org/

[16] [n. d.]. Team Cymru. ([n. d.]). Retrieved 15-08-2016 from http://www.team-cymru.org/

[17] [n. d.]. The Internet Corporation for Assigned Names and Numbers (ICANN). ([n. d.]). Retrieved 10-04-2017 from https://www.icann.org/

[18] [n. d.]. URLVoid: Website Reputation Checker Tool. ([n. d.]). Retrieved 19-07-2016 from http://www.urlvoid.com/

[19] [n. d.]. Web of Trust (WOT) – Crowdsourced web safety. ([n. d.]). Retrieved 19-07-2016 from https://www.mywot.com/

[20] [n. d.]. Who.is: WHOIS Search, Domain Name, Website, and IP Tools. ([n. d.]). Retrieved 12-04-2017 from https://who.is

Table 4. A Summary of the Proposed Categorization Scheme and Challenges

Realm	Component	Dimension	Challenge
Data Sources	DNS Data	*1. Where are the Data Collected* a) Host-resolver b) DNS-DNS *2. How are the Data Collected* a) Active b) Passive	1. Hard to obtain access to DNS data 2. Hard to share data to run comparative tests
	Data Enrichment	*1. Type of the Enrichment Data* a) Geo-location b) ASN c) Registration records d) IP/domain black-/whitelists e) Associated resource records f) Network information	1. Historical change of the enrichment information 2. Enrichment information management 3. Limited or payable access to the information
	Ground Truth	*1. Type of the Ground Truth* a) Malicious b) Benign	1. Low-quality public blacklists and mixed content 2. Low-quality/non-representative benign domains 3. Inconsistent domain levels and proprietary data 4. Imbalanced datasets
Approaches	Features	*1. Internal vs. Contextual* a) Internal b) Contextual *2. DNS Dataset Dependent vs. DNS Dataset Independent* a) Dependent b) Independent *3. Mono Domain vs. Multi Domains* a) Mono b) Multi	1. Easiness to forge or to manipulate 2. Reproducibility of the results 3. Data access, data sharing
	Detection Methods	*1. Knowledge Based vs. Machine Learning Based* a) Knowledge based b) Machine learning based 1) Supervised learning 2) Semi-supervised learning 3) Unsupervised learning c) Hybrid approaches	1. Feature resilience evaluation 2. Performance and overhead 3. Ability to work in real-time 4. Adaptiveness of adversaries 5. Systematic comparison of techniques
	Outcome	*1. Malicious Behavior Agnostic vs. Malicious Behavior Specific* a) Agnostic b) Specific	1. Difficulty to compare approaches aiming at different goals 2. No clear definition what domains are malicious
Evaluation	Metrics	*1. Metric Types* a) TPR/Recall b) FPR c) TNR d) FNR e) Precision f) Accuracy g) F1-score h) AUC	1. Lack of well agreed upon metrics 2. Lack of problem-specific metric
	Evaluation Strategies	*1. Evaluation Types* a) Whole dataset b) One round train-test split c) Leave-p-out cross-validation d) K-fold cross-validation e) Cross-networks validation f) Cross-time validation g) Cross-blacklists validation	1. New knowledge validation 2. Lack of public reference datasets 3. Unified metrics calculation

[21] [n. d.]. Yahoo Webspam Database. ([n. d.]). http://barcelona.research.yahoo.net/webspam/datasets/uk2007/ Service discontinued.

[22] 2002. Vulnerability Note VU#457875: Various DNS Service Implementations Generate Multiple Simultaneous Queries for the Same Resource Record. (November 2002). Retrieved 07-11-2016 from http://www.kb.cert.org/vuls/id/457875

[23] 2008. Vulnerability Note VU#800113: Multiple DNS Implementations Vulnerable to Cache Poisoning. (July 2008). Retrieved 07-11-2016 from http://www.kb.cert.org/vuls/id/800113

[24] Pieter Agten, Wouter Joosen, Frank Piessens, and Nick Nikiforakis. 2015. Seven Months' Worth of Mistakes: A Longitudinal Study of Typosquatting Abuse. In *Proceedings of the Network and Distributed System Security Symposium.*

[25] Alexa. [n. d.]. Alexa Top Sites. ([n. d.]). Retrieved 30-03-2016 from http://aws.amazon.com/alexa-top-sites/

[26] Kamal Alieyan, Ammar ALmomani, Ahmad Manasrah, and Mohammed M. Kadhum. 2015. A Survey of Botnet Detection Based on DNS. *Neural Computing and Applications* (2015), 1–18.

[27] Hyrum S. Anderson, Jonathan Woodbridge, and Bobby Filar. 2016. DeepDGA: Adversarially-Tuned Domain Generation and Detection. In *Proceedings of the ACM Workshop on Artificial Intelligence and Security.* 13–21.

[28] Manos Antonakakis, Roberto Perdisci, David Dagon, Wenke Lee, and Nick Feamster. 2010. Building a Dynamic Reputation System for DNS. In *Proceedings of the USENIX Security Symposium.* 273–290.

[29] Manos Antonakakis, Roberto Perdisci, Wenke Lee, Nikolaos Vasiloglou, II, and David Dagon. 2011. Detecting Malware Domains at the Upper DNS Hierarchy. In *Proceedings of the USENIX Security Symposium.* 27–27.

[30] Manos Antonakakis, Roberto Perdisci, Yacin Nadji, Nikolaos Vasiloglou, Saeed Abu-Nimeh, Wenke Lee, and David Dagon. 2012. From Throw-away Traffic to Bots: Detecting the Rise of DGA-based Malware. In *Proceedings of the USENIX Security Symposium.* 24–24.

[31] Ionut Arghire. 2016. Sarvdap Spambot Checks IP Blacklists. (October 2016). Retrieved 08-11-2016 from http://www.securityweek.com/sarvdap-spambot-checks-ip-blacklists

[32] Anirban Banerjee, Md Sazzadur Rahman, and Michalis Faloutsos. 2011. SUT: Quantifying and Mitigating URL Typosquatting. *Computer Networks* 55, 13 (2011), 3001 – 3014.

[33] Steven M. Bellovin. 1995. Using the Domain Name System for System Break-ins. In *Proceedings of the Conference on USENIX UNIX Security Symposium - Volume 5.* 18–18.

[34] Andreas Berger and Wilfried N. Gansterer. 2013. Modeling DNS Agility with DNSMap. In *Proceedings of the IEEE Conference on Computer Communications Workshops.* 387–392.

[35] Leyla Bilge, Engin Kirda, Christopher Kruegel, and Marco Balduzzi. 2011. EXPOSURE: Finding Malicious Domains Using Passive DNS Analysis. In *Proceedings of the Network and Distributed System Security Symposium.*

[36] Leyla Bilge, Sevil Sen, Davide Balzarotti, Engin Kirda, and Christopher Kruegel. 2014. Exposure: A Passive DNS Analysis Service to Detect and Report Malicious Domains. *ACM Transactions on Information and System Security* 16, 4 (apr 2014), 14:1–14:28.

[37] N. Brownlee, K. C. Claffy, and E. Nemeth. 2001. DNS Measurements at a Root Server. In *Proceedings of the Global Telecommunications Conference*, Vol. 3. 1672–1676.

[38] Davide Canali, Marco Cova, Giovanni Vigna, and Christopher Kruegel. 2011. Prophiler: A Fast Filter for the Large-scale Detection of Malicious Web Pages. In *Proceedings of the International Conference on World Wide Web.* 197–206.

[39] Biz Carson. 2016. This Guy Bought 'Google.com' from Google for One Minute. (September 2016). Retrieved 10-01-2017 from http://www.businessinsider.com/this-guy-bought-googlecom-from-google-for-one-minute-2015-9

[40] Sebastian Castro, Duane Wessels, Marina Fomenkov, and Kimberly Claffy. 2008. A Day at the Root of the Internet. *SIGCOMM Computer Communication Review* 38, 5 (2008), 41–46.

[41] Olivier Chapelle, Bernhard Schlkopf, and Alexander Zien. 2010. *Semi-Supervised Learning* (1st ed.). The MIT Press.

[42] Nitesh V. Chawla. 2005. *Data Mining for Imbalanced Datasets: An Overview.* 853–867.

[43] Pern Hui Chia and Svein Johan Knapskog. 2012. Re-evaluating the Wisdom of Crowds in Assessing Web Security. In *Proceedings of the International Conference on Financial Cryptography and Data Security.* 299–314.

[44] Daiki Chiba, Takeshi Yagi, Mitsuaki Akiyama, Toshiki Shibahara, Takeshi Yada, Tatsuya Mori, and Shigeki Goto. 2016. DomainProfiler: Discovering Domain Names Abused in Future. *Proceedings of the Annual IEEE/IFIP International Conference on Dependable Systems and Networks* (2016), 491–502.

[45] Hyunsang Choi and Heejo Lee. 2012. Identifying Botnets by Capturing Group Activities in DNS Traffic. *Computer Networks* 56, 1 (2012), 20 – 33.

[46] Hyunsang Choi, Heejo Lee, and Hyogon Kim. 2009. BotGAD: Detecting Botnets by Capturing Group Activities in Network Traffic. In *Proceedings of the International ICST Conference on COMmunication System softWAre and middlewaRE.* 2:1–2:8.

[47] Hyunsang Choi, Hanwoo Lee, Heejo Lee, and Hyogon Kim. 2007. Botnet Detection by Monitoring Group Activities in DNS Traffic. In *Proceedings of the IEEE International Conference on Computer and Information Technology.* 715–720.

[48] Marco Cova, Christopher Kruegel, and Giovanni Vigna. 2010. Detection and Analysis of Drive-by-download Attacks and Malicious JavaScript Code. In *Proceedings of the International Conference on World Wide Web.* 281–290.

[49] L. Daigle. 2004. *WHOIS Protocol Specification.* RFC 3912. Internet Engineering Task Force. https://tools.ietf.org/html/rfc3912

[50] Ruchi Dhole and Shobha Lolge. 2016. A Survey of Botnet Detection Techniques and Research Challenges. *International Journal of Innovative Research in Computer and Communication Engineering* 4 (2016), 244–249. Issue 1.

[51] Christian J. Dietrich and Christian Rossow. 2009. Empirical Research of IP Blacklists. In *Proceedings of the Information Security Solutions Europe Conference*. 163–171.

[52] Christian J. Dietrich, Christian Rossow, Felix C. Freiling, Herbert Bos, Maarten van Steen, and Norbert Pohlmann. 2011. On Botnets That Use DNS for Command and Control. In *Proceedings of the European Conference on Computer Network Defense*. 9–16.

[53] Zakir Durumeric, David Adrian, Ariana Mirian, Michael Bailey, and J. Alex Halderman. 2015. A Search Engine Backed by Internet-Wide Scanning. In *Proceedings of the ACM Conference on Computer and Communications Security*.

[54] Birhanu Eshete, Adolfo Villafiorita, and Komminist Weldemariam. 2013. BINSPECT: Holistic Analysis and Detection of Malicious Web Pages. In *Proceedings of the International ICST Conference on Security and Privacy in Communication Networks*. 149–166.

[55] Farsight Security, Inc. [n. d.]. DNS Database. ([n. d.]). Retrieved 28-03-2016 from https://www.dnsdb.info/

[56] Maryam Feily, Alireza Shahrestani, and Sureswaran Ramadass. 2009. A Survey of Botnet and Botnet Detection. In *Proceedings of the International Conference on Emerging Security Information, Systems and Technologies*. 268–273.

[57] Mark Felegyhazi, Christian Kreibich, and Vern Paxson. 2010. On the Potential of Proactive Domain Blacklisting. In *Proceedings of the USENIX Conference on Large-scale Exploits and Emergent Threats*.

[58] Paul S. Ferrell. [n. d.]. Apt Infection Discovery Using DNS Data. ([n. d.]). Retrieved 10-08-2016 from http://permalink.lanl.gov/object/view?what=info:lanl-repo/lareport/LA-UR-13-23109

[59] Sean Ford, Marco Cova, Christopher Kruegel, and Giovanni Vigna. 2009. Wepawet. (2009). http://wepawet.iseclab.org/ Service discontinued.

[60] Y. Fu, L. Yu, O. Hambolu, I. Ozcelik, B. Husain, J. Sun, K. Sapra, D. Du, C. T. Beasley, and R. R. Brooks. 2017. Stealthy Domain Generation Algorithms. *IEEE Transactions on Information Forensics and Security* 12, 6 (2017), 1430–1443.

[61] Kensuke Fukuda and John Heidemann. 2015. Detecting Malicious Activity with DNS Backscatter. In *Proceedings of the ACM SIGCOMM Internet Measurement Conference*. 197–210.

[62] Hongyu Gao, Vinod Yegneswaran, Yan Chen, Phillip Porras, Shalini Ghosh, Jian Jiang, and Haixin Duan. 2013. An Empirical Reexamination of Global DNS Behavior. In *ACM SIGCOMM Computer Communication Review*. 267–278.

[63] H. Gao, V. Yegneswaran, J. Jiang, Y. Chen, P. Porras, S. Ghosh, and H. Duan. 2016. Reexamining DNS From a Global Recursive Resolver Perspective. *IEEE/ACM Transactions on Networking* 24, 1 (Feb 2016), 43–57.

[64] M. Grill, I. Nikolaev, V. Valeros, and M. Rehak. 2015. Detecting DGA Malware using NetFlow. In *Proceedings of the IFIP/IEEE International Symposium on Integrated Network Management*. 1304–1309.

[65] H. Guerid, K. Mittig, and A. Serhrouchni. 2013. Privacy-preserving Domain-flux Botnet Detection in a Large Scale Network. In *Proceedings of the International Conference on Communication Systems and Networks*. 1–9.

[66] Fariba Haddadi, H. Gunes Kayacik, A. Nur Zincir-Heywood, and Malcolm I. Heywood. 2013. Malicious Automatically Generated Domain Name Detection Using Stateful-SBB. In *Proceedings of the European Conference Applications of Evolutionary Computation*. 529–539.

[67] F. Haddadi and A.N. Zincir-Heywood. 2013. Analyzing String Format-based Classifiers for Botnet Detection: GP and SVM. In *Proceedings of the IEEE Congress on Evolutionary Computation*. 2626–2633.

[68] Shuang Hao, Nick Feamster, and Ramakant Pandrangi. 2011. Monitoring the Initial DNS Behavior of Malicious Domains. In *Proceedings of the ACM SIGCOMM Internet Measurement Conference*. 269–278.

[69] Shuang Hao, Alex Kantchelian, Brad Miller, Vern Paxson, and Nick Feamster. 2016. PREDATOR: Proactive Recognition and Elimination of Domain Abuse at Time-Of-Registration. In *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security*. 1568–1579.

[70] Shuang Hao, Matthew Thomas, Vern Paxson, Nick Feamster, Christian Kreibich, Chris Grier, and Scott Hollenbeck. 2013. Understanding the Domain Registration Behavior of Spammers. In *Proceedings of the ACM SIGCOMM Internet Measurement Conference (IMC '13)*. 63–76.

[71] Dominik Herrmann, Christian Banse, and Hannes Federrath. 2013. Behavior-based Tracking: Exploiting Characteristic Patterns in DNS Traffic. *Computers & Security* 39, Part A (2013), 17 – 33.

[72] Hoglund, Greg and McGraw, Gary. 2004. *Exploiting Software: How to Break Code.* Pearson Higher Education.

[73] Thorsten Holz, Christian Gorecki, Konrad Rieck, and Felix C. Freiling. 2008. Measuring and Detecting Fast-Flux Service Networks. In *Proceedings of the Network and Distributed System Security Symposium*.

[74] Ching-Hsiang Hsu, Chun-Ying Huang, and Kuan-Ta Chen. 2010. Fast-flux Bot Detection in Real Time. In *Proceedings of the International Conference on Recent Advances in Intrusion Detection*. 464–483.

[75] Xin Hu, M. Knysz, and K.G. Shin. 2011. Measurement and Analysis of Global IP-usage Patterns of Fast-flux Botnets. In *Proceedings of the IEEE INFOCOM*. 2633–2641.

[76] Xin Hu, Matthew Knysz, and Kang G Shin. 2009. RB-Seeker: Auto-detection of Redirection Botnets. In *Proceedings of the Network and Distributed System Security Symposium*.

[77] Yonghong Huang and P. Greve. 2015. Large Scale Graph Mining for Web Reputation Inference. In *Proceedings of the IEEE International Workshop on Machine Learning for Signal Processing*. 1–6.

[78] A. K. Jain, M. N. Murty, and P. J. Flynn. 1999. Data Clustering: A Review. *Comput. Surveys* 31, 3 (Sept. 1999), 264–323.

[79] Nan Jiang, Jin Cao, Yu Jin, Li Li, and Zhi-Li Zhang. 2010. Identifying Suspicious Activities through DNS Failure Graph Analysis. In *Proceedings of the IEEE International Conference on Network Protocols*. 144–153.

[80] Alexandros Kapravelos, Marco Cova, Christopher Kruegel, and Giovanni Vigna. 2011. Escape from Monkey Island: Evading High-Interaction Honeyclients. In *Proceedings of the International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment*. 124–143.

[81] A. M. Kara, H. Binsalleeh, M. Mannan, A. Youssef, and M. Debbabi. 2014. Detection of Malicious Payload Distribution Channels in DNS. In *Proceedings of the IEEE International Conference on Communications*. 853–858.

[82] Issa M. Khalil, Ting Yu, and Bei Guan. 2016. Discovering Malicious Domains through Passive DNS Data Graph Analysis. In *Proceedings of the ACM Symposium on Information, Computer and Communications Security*. 663–674.

[83] M. T. Khan, X. Huo, Z. Li, and C. Kanich. 2015. Every Second Counts: Quantifying the Negative Externalities of Cybercrime via Typosquatting. In *Proceedings of the IEEE Symposium on Security and Privacy*. 135–150.

[84] S. Khattak, N. R. Ramay, K. R. Khan, A. A. Syed, and S. A. Khayam. 2014. A Taxonomy of Botnet Behavior, Detection, and Defense. *IEEE Communications Surveys & Tutorials* 16, 2 (2014), 898–924.

[85] Nizar Kheir, Frédéric Tran, Pierre Caron, and Nicolas Deschamps. 2014. Mentor: Positive DNS Reputation to Skim-Off Benign Domains in Botnet C&C Blacklists. In *Proceedings of the IFIP TC International Conference on ICT Systems Security and Privacy Protection*. 1–14.

[86] M. Khonji, Y. Iraqi, and A. Jones. 2013. Phishing Detection: A Literature Survey. *IEEE Communications Surveys and Tutorials* 15, 4 (2013), 2091–2121.

[87] Panagiotis Kintis, Najmeh Miramirkhani, Charles Lever, Yizheng Chen, Rosa Romero-Gómez, Nikolaos Pitropakis, Nick Nikiforakis, and Manos Antonakakis. 2017. Hiding in Plain Sight: A Longitudinal Study of Combosquatting Abuse. In *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security*. 569–586.

[88] Matthias Kirchler, Dominik Herrmann, Jens Lindemann, and Marius Kloft. 2016. Tracked Without a Trace: Linking Sessions of Users by Unsupervised Learning of Patterns in Their DNS Traffic. In *Proceedings of the ACM Workshop on Artificial Intelligence and Security*. 23–34.

[89] Maria Konte, Nick Feamster, and Jaeyeon Jung. 2009. Dynamics of Online Scam Hosting Infrastructure. In *Proceedings of the International Conference on Passive and Active Network Measurement*. 219–228.

[90] S. B. Kotsiantis. 2007. Supervised Machine Learning: A Review of Classification Techniques. In *Proceedings of the Conference on Emerging Artificial Intelligence Applications in Computer Engineering*. 3–24.

[91] Athanasios Kountouras, Panagiotis Kintis, Charles Lever, Yizheng Chen, Yacin Nadji, David Dagon, and Manos Antonakakis. 2016. Enabling Network Security Through Active DNS Datasets. In *Proceedings of the International Symposium on Research in Attacks, Intrusions, and Defenses*. 188–208.

[92] David Kravets. 2008. ICANN and IANA Sites Hacked, Redirected. (June 2008). Retrieved 07-11-2016 from https://www.wired.com/2008/06/icann-and-iana/

[93] Srinivas Krishnan, Teryl Taylor, Fabian Monrose, and John McHugh. 2013. Crossing the Threshold: Detecting Network Malfeasance via Sequential Hypothesis Testing. In *Proceedings of the Annual IEEE/IFIP International Conference on Dependable Systems and Networks*. 1–12.

[94] Marc Kührer and Thorsten Holz. 2012. An Empirical Analysis of Malware Blacklists. *Praxis der Informationsverarbeitung und Kommunikation* 35, 1 (2012), 11–16.

[95] Marc Kührer, Christian Rossow, and Thorsten Holz. 2014. Paint it Black: Evaluating the Effectiveness of Malware Blacklists. In *Proceedings of the International Symposium on Research in Attacks, Intrusions and Defenses*.

[96] Jonghoon Kwon, Jeongsik Kim, Jehyun Lee, Heejo Lee, and Adrian Perrig. 2014. PsyBoG: Power Spectral Density Analysis for Detecting Botnet Groups. In *Proceedings of the International Conference on Malicious and Unwanted Software*. 85–92.

[97] Kyle York. 2016. Dyn Statement on 10/21/2016 DDoS Attack. (October 2016). Retrieved 28-10-2016 from http://dyn.com/blog/dyn-statement-on-10212016-ddos-attack/

[98] Tobias Lauinger, Kaan Onarlioglu, Abdelberi Chaabane, William Robertson, and Engin Kirda. 2016. WHOIS Lost in Translation: (Mis)Understanding Domain Name Expiration and Re-Registration. In *Proceedings of the ACM SIGCOMM Internet Measurement Conference*. 247–253.

[99] Felix Leder and Tillmann Werner. 2009. Know Your Enemy: Containing Conficker. (2009). Retrieved 21-05-2016 from https://www.honeynet.org/files/KYE-Conficker.pdf

[100] J. Lee, J. Kwon, H. J. Shin, and H. Lee. 2010. Tracking Multiple C&C Botnets by Analyzing DNS Traffic. In *Proceedings of the IEEE Workshop on Secure Network Protocols*. 67–72.

[101] J. Lee and H. Lee. 2014. GMAD: Graph-based malware activity detection by DNS traffic analysis. *Computer Communications* 49 (2014), 33–47.

[102] C. Lever, R. Walls, Y. Nadji, D. Dagon, P. McDaniel, and M. Antonakakis. 2016. Domain-Z: 28 Registrations Later Measuring the Exploitation of Residual Trust in Domains. In *Proceedings of the IEEE Symposium on Security and Privacy*.

691–706.

[103] Z. Li, S. Alrwais, X. Wang, and E. Alowaisheq. 2014. Hunting the Red Fox Online: Understanding and Detection of Mass Redirect-Script Injections. In *Proceedings of the IEEE Symposium on Security and Privacy*. 3–18.

[104] Daiping Liu, Shuai Hao, and Haining Wang. 2016. All Your DNS Records Point to Us: Understanding the Security Threats of Dangling DNS Records. In *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security*. 1414–1425.

[105] Justin Ma, Lawrence K. Saul, Stefan Savage, and Geoffrey M. Voelker. 2009. Beyond Blacklists: Learning to Detect Malicious Web Sites from Suspicious URLs. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 1245–1254.

[106] Justin Ma, Lawrence K. Saul, Stefan Savage, and Geoffrey M. Voelker. 2011. Learning to Detect Malicious URLs. *ACM Transactions on Intelligent Systems and Technology* 2, 3 (2011), 30:1–30:24.

[107] X. Ma, J. Zhang, J. Tao, J. Li, J. Tian, and X. Guan. 2014. DNSRadar: Outsourcing Malicious Domain Detection Based on Distributed Cache-Footprints. *IEEE Transactions on Information Forensics and Security* 9, 11 (Nov 2014), 1906–1921.

[108] D. Mahjoub. 2013. Monitoring a Fast Flux Botnet using Recursive and Passive DNS: A Case Study. In *Proceedings of the eCrime Researchers Summit*. 1–9.

[109] Pratyusa Manadhata, Sandeep Yadav, Prasad Rao, and William Horne. 2014. Detecting Malicious Domains via Graph Inference. In *Proceedings of the European Symposium on Research in Computer Security*. 1–18.

[110] Samuel Marchal, Jérôme François, Radu State, and Thomas Engel. 2012. Proactive Discovery of Phishing Related Domain Names. In *Proceedings of the International Symposium Research in Attacks, Intrusions, and Defenses*. 190–209.

[111] MaxMind. [n. d.]. GeoLite2 Databases. ([n. d.]). Retrieved 28-04-2016 from http://www.maxmind.com

[112] Igor Mishsky, Nurit Gal-Oz, and Ehud Gudes. 2015. A Topology Based Flow Model for Computing Domain Reputation. *Proceedings of the Annual IFIP WG 11.3 Working Conference on Data and Applications Security and Privacy* (2015), 277–292.

[113] Thomas M. Mitchell. 1997. *Machine Learning* (1 ed.). McGraw-Hill, Inc.

[114] P. Mockapetris. 1983. *Domain Names: Concepts and Facilities*. Technical Report. Internet Engineering Task Force. https://tools.ietf.org/html/rfc882

[115] P.V. Mockapetris. 1983. *Domain Names: Implementation and Specification*. Technical Report. Internet Engineering Task Force. https://tools.ietf.org/html/rfc883

[116] P. Mockapetris. 1987. *Domain Names: Concepts and Facilities*. Technical Report. Internet Engineering Task Force. https://tools.ietf.org/html/rfc1034

[117] P. Mockapetris. 1987. *Domain Names: Implementation and Specification*. Technical Report. Internet Engineering Task Force. https://tools.ietf.org/html/rfc1035

[118] J. A. Morales, A. Al-Bataineh, Shouhuai Xu, and R. Sandhu. 2009. Analyzing DNS Activities of Bot Processes. In *Proceedings of the International Conference on Malicious and Unwanted Software*. 98–103.

[119] Jason Murdock. 2016. Linux Mint Hack: 71,000 User Accounts Stolen and Malware Planted using Tsunami Backdoor. (February 2016). http://goo.gl/CNY4gB

[120] Yacin Nadji, Manos Antonakakis, Roberto Perdisci, and Wenke Lee. 2013. Connected Colors: Unveiling the Structure of Criminal Networks. In *Proceedings of the International Symposium on Research in Attacks, Intrusions, and Defenses*. 390–410.

[121] Jose Nazario and Thorsten Holz. 2008. As the Net Churns: Fast-flux Botnet Observations. In *Proceedings of the International Conference on Malicious and Unwanted Software*. 24–31.

[122] Terry Nelms, Roberto Perdisci, and Mustaque Ahamad. 2013. ExecScent: Mining for New C&C Domains in Live Networks with Adaptive Control Protocol Templates. In *Proceedings of the USENIX Security Symposium*. 589–604.

[123] Nick Nikiforakis, Marco Balduzzi, Lieven Desmet, Frank Piessens, and Wouter Joosen. 2014. Soundsquatting: Uncovering the Use of Homophones in Domain Squatting. In *Proceedings of the International Conference on Information Security*. 291–308.

[124] Nick Nikiforakis, Steven Van Acker, Wannes Meert, Lieven Desmet, Frank Piessens, and Wouter Joosen. 2013. Bitsquatting: Exploiting Bit-flips for Fun, or Profit?. In *Proceedings of the International Conference on World Wide Web*. 989–998.

[125] OpenDNS. [n. d.]. PhishTank. ([n. d.]). Retrieved 16-05-2016 from https://www.phishtank.com/

[126] A. Oprea, Z. Li, T. F. Yen, S. H. Chin, and S. Alrwais. 2015. Detection of Early-Stage Enterprise Infection by Mining Large-Scale Log Data. In *Proceedings of the Annual IEEE/IFIP International Conference on Dependable Systems and Networks*. 45–56.

[127] Emanuele Passerini, Roberto Paleari, Lorenzo Martignoni, and Danilo Bruschi. 2008. FluXOR: Detecting and Monitoring Fast-Flux Service Networks. In *Proceedings of the International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment*. 186–206.

[128] R. Perdisci, I. Corona, D. Dagon, and Wenke Lee. 2009. Detecting Malicious Flux Service Networks through Passive Analysis of Recursive DNS Traces. In *Proceedings of the Annual Computer Security Applications Conference*. 311–320.

[129] R. Perdisci, I. Corona, and G. Giacinto. 2012. Early Detection of Malicious Flux Networks via Large-Scale Passive DNS Traffic Analysis. *IEEE Transactions on Dependable and Secure Computing* 9, 5 (2012), 714–726.

[130] Iria Prieto, Eduardo Magaña, Daniel Morató, and Mikel Izal. 2011. Botnet Detection based on DNS Records and Active Probing. In *Proceedings of the International Conference on Security and Cryptography*. 307–316.

[131] Niels Provos, Panayiotis Mavrommatis, Moheeb Abu Rajab, and Fabian Monrose. 2008. All Your iFRAMEs Point to Us. In *Proceedings of the USENIX Security Symposium*. 1–15.

[132] Zhiyun Qian, Zhuoqing Morley Mao, Yinglian Xie, and Fang Yu. 2010. On Network-level Clusters for Spam Detection. In *Proceedings of the Network and Distributed System Security Symposium*.

[133] B. Rahbarinia, R. Perdisci, and M. Antonakakis. 2015. Segugio: Efficient Behavior-Based Tracking of Malware-Control Domains in Large ISP Networks. In *Proceedings of the Annual IEEE/IFIP International Conference on Dependable Systems and Networks*. 403–414.

[134] Babak Rahbarinia, Roberto Perdisci, and Manos Antonakakis. 2016. Efficient and Accurate Behavior-Based Tracking of Malware-Control Domains in Large ISP Networks. *ACM Transactions on Privacy and Security* 19, 2 (Aug. 2016), 4:1–4:31.

[135] A. Ramachandran, D. Dagon, and Nick Feamster. 2006. Can DNS-based Blacklists Keep Up with Bots. In *Proceedings of the Conference on Email and Anti-Spam*.

[136] Christian Rossow. 2014. Amplification Hell: Revisiting Network Protocols for DDoS Abuse. In *Proceedings of the Network and Distributed System Security Symposium*.

[137] Doyen Sahoo, Chenghao Liu, and Steven C. H. Hoi. 2017. Malicious URL Detection using Machine Learning: A Survey. *CoRR* abs/1701.07179 (2017). http://arxiv.org/abs/1701.07179

[138] Arthur L Samuel. 1959. Some Studies in Machine Learning using the Game of Checkers. *IBM Journal of research and development* 3, 3 (1959), 210–229.

[139] Kazumichi Sato, Keisuke Ishibashi, Tsuyoshi Toyono, and Nobuhisa Miyake. 2010. Extending Black Domain Name List by Using Co-occurrence Relation Between DNS Queries. In *Proceedings of the 3rd USENIX Conference on Large-scale Exploits and Emergent Threats: Botnets, Spyware, Worms, and More*. 8–8.

[140] Stefano Schiavoni, Federico Maggi, Lorenzo Cavallaro, and Stefano Zanero. 2014. Phoenix: DGA-Based Botnet Tracking and Intelligence. In *Proceedings of the International Conference Detection of Intrusions and Malware, and Vulnerability Assessment*. 192–211.

[141] Security and Stability Advisory Committee. 2009. SAC 40: Measures to Protect Domain Registration Services Against Exploitation or Misuse. (August 2009). Retrieved 07-11-2016 from https://www.icann.org/en/system/files/files/sac-040-en.pdf

[142] C. Seifert, I. Welch, P. Komisarczuk, C. U. Aval, and B. Endicott-Popovsky. 2008. Identification of Malicious Web Pages through Analysis of Underlying DNS and Web Server Relationships. In *Proceedings of the IEEE Conference on Local Computer Networks*. 935–941.

[143] Steve Sheng, Brad Wardman, Gary Warner, Lorrie Faith Cranor, Jason Hong, and Chengshan Zhang. 2009. An Empirical Analysis of Phishing Blacklists. In *Proceedings of the Conference on Email and Anti-Spam*.

[144] Seungwon Shin, Zhaoyan Xu, and Guofei Gu. 2012. EFFORT: Efficient and Effective Bot Malware Detection. In *Proceedings of the IEEE INFOCOM*. 2846–2850.

[145] S. Sinha, M. Bailey, and F. Jahanian. 2008. Shades of Grey: On the Effectiveness of Reputation-based "Blacklists". In *Proceedings of the International Conference on Malicious and Unwanted Software*. 57–64.

[146] A. K. Sood and S. Zeadally. 2016. A Taxonomy of Domain-Generation Algorithms. *IEEE Security & Privacy* 14, 4 (2016), 46–53.

[147] Nikita Spirin and Jiawei Han. 2012. Survey on Web Spam Detection: Principles and Algorithms. *ACM SIGKDD Explorations Newsletter* 13, 2 (2012), 50–64.

[148] Etienne Stalmans. 2011. A Framework for DNS Based Detection and Mitigation of Malware Infections on a Network. In *Proceedings of the Information Security South Africa Conference*.

[149] Matija Stevanovic, Jens Myrup Pedersen, Alessandro D'Alconzo, and Stefan Ruehrup. 2017. A Method for Identifying Compromised Clients Based on DNS Traffic Analysis. *International Journal of Information Security* 16, 2 (2017), 115–132.

[150] Matija Stevanovic, Jens Myrup Pedersen, Alessandro D'Alconzo, Stefan Ruehrup, and Andreas Berger. 2015. On the Ground Truth Problem of Malicious DNS Traffic Analysis. *Computers & Security* 55 (nov 2015), 142–158.

[151] Elizabeth Stinson and John C. Mitchell. 2008. Towards Systematic Evaluation of the Evadability of Bot/Botnet Detection Methods. In *Proceedings of the USENIX Workshop on Offensive Technologies*. 5:1–5:9.

[152] Janos Szurdi, Balazs Kocso, Gabor Cseh, Jonathan Spring, Mark Felegyhazi, and Chris Kanich. 2014. The Long "Taile" of Typosquatting Domain Names. In *Proceedings of the USENIX Security Symposium*. 191–206.

[153] The DNS-BH project. [n. d.]. DNS-BH – Malware Domain Blocklist. ([n. d.]). Retrieved 16-05-2016 from http://www.malwaredomains.com/

[154] Matthew Thomas and Aziz Mohaisen. 2014. Kindred Domains: Detecting and Clustering Botnet Domains Using DNS Traffic. In *Proceedings of the International Conference on World Wide Web*. 707–712.

[155] Van Tong and Giang Nguyen. 2016. A Method for Detecting DGA Botnet Based on Semantic and Cluster Analysis. In *Proceedings of the Symposium on Information and Communication Technology*. 272–277.

[156] Verisign, Inc. 2016. Internet Grows to 314 Million Domain Names in the Fourth Quarter of 2015. (April 2016). Retrieved 28-04-2016 from https://www.verisign.com/assets/press-release-DNIB-april2016.pdf

[157] R. Villamarin-Salomon and J. C. Brustoloni. 2008. Identifying Botnets Using Anomaly Detection Techniques Applied to DNS Traffic. In *Proceedings of the IEEE Consumer Communications and Networking Conference*. 476–481.

[158] Ricardo Villamarín-Salomón and José Carlos Brustoloni. 2009. Bayesian Bot Detection Based on DNS Traffic Similarity. In *Proceedings of the ACM Symposium on Applied Computing*. 2035–2041.

[159] VirusTotal, Subsidiary of Google. [n. d.]. VirusTotal – Free Online Virus, Malware and URL Scanner. ([n. d.]). Retrieved 04-05-2016 from https://www.virustotal.com/

[160] Qiong Wei and Roland L. Dunbrack, Jr. 2013. The Role of Balanced Training and Testing Data Sets for Binary Classifiers in Bioinformatics. *PLOS ONE* 8 (07 2013), 1–12.

[161] Florian Weimer. 2005. Passive DNS Replication. In *Proceedings of the Conference on Computer Security Incident*. 98.

[162] Gilbert Wondracek, Thorsten Holz, Christian Platzer, Engin Kirda, and Christopher Kruegel. 2010. Is the Internet for Porn? An Insight Into the Online Adult Industry. In *Proceedings of the Annual Workshop on the Economics of Information Security*.

[163] Jonathan Woodbridge, Hyrum S. Anderson, Anjum Ahuja, and Daniel Grant. 2016. Predicting Domain Generation Algorithms with Long Short-Term Memory Networks. (2016). arXiv:arXiv:1611.00791

[164] Steven Wright. 2012. Cybersquatting at the Intersection of Internet Domain Names and Trademark Law. *IEEE Communications Surveys Tutorials* 14, 1 (2012), 193–205.

[165] Sandeep Yadav, Ashwath Kumar Krishna Reddy, A.L. Narasimha Reddy, and Supranamaya Ranjan. 2010. Detecting Algorithmically Generated Malicious Domain Names. In *Proceedings of the ACM SIGCOMM Conference on Internet Measurement*. 48–61.

[166] Sandeep Yadav, Ashwath Kumar Krishna Reddy, A.L. Narasimha Reddy, and Supranamaya Ranjan. 2012. Detecting Algorithmically Generated Domain-Flux Attacks With DNS Traffic Analysis. *IEEE/ACM Transactions on Networking* 20, 5 (2012), 1663–1677.

[167] Sandeep Yadav and A. L. Narasimha Reddy. 2011. Winning with DNS Failures: Strategies for Faster Botnet Detection. In *Proceedings of the International ICST Conference Security and Privacy in Communication Networks*. 446–459.

[168] Ting-Fang Yen and Michael K. Reiter. 2008. Traffic Aggregation for Malware Detection. In *Proceedings of the International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment*. 207–227.

[169] Bin Yu, Les Smith, and Mark Threefoot. 2014. Semi-supervised Time Series Modeling for Real-Time Flux Domain Detection on Passive DNS Traffic. In *Proceedings of the International Conference Machine Learning and Data Mining in Pattern Recognition*. 258–271.

[170] Bojan Zdrnja, Nevil Brownlee, and Duane Wessels. 2007. Passive Monitoring of DNS Anomalies. In *Proceedings of the International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment*. 129–139.

[171] Jialong Zhang, Sabyasachi Saha, Guofei Gu, Sung-Ju Lee, and Marco Mellia. 2015. Systematic Mining of Associated Server Herds for Malware Campaign Discovery. In *Proceedings of the IEEE International Conference on Distributed Computing Systems*. 630–641.

[172] F. Zhao, Y. Hori, and K. Sakurai. 2007. Analysis of Privacy Disclosure in DNS Query. In *Proceedings of the International Conference on Multimedia and Ubiquitous Engineering*. 952–957.

[173] Xiaojin Zhu. 2005. *Semi-Supervised Learning Literature Survey.* Technical Report 1530. Computer Science, University of Wisconsin-Madison.

[174] Futai Zou, Siyu Zhang, Weixiong Rao, and Ping Yi. 2015. Detecting Malware Based on DNS Graph Mining. *International Journal of Distributed Sensor Networks* 2015 (2015).

[175] Hiba Zuhair, Ali Selamat, and Mazleena Salleh. 2016. Feature Selection for Phishing Detection: A Review of Research. *International Journal of Intelligent Systems Technologies and Applications* 15, 2 (May 2016), 147–162.