

## Article

# Botnet Detection Technology Based on DNS

Xingguo Li <sup>1</sup>, Junfeng Wang <sup>2,\*</sup> and Xiaosong Zhang <sup>3</sup>

<sup>1</sup> National Key Laboratory of Fundamental Science on Synthetic Vision, Sichuan University, Chengdu 610065, China; xglee@scu.edu.cn

<sup>2</sup> College of Computer Science, Sichuan University, Chengdu 610065, China

<sup>3</sup> Center for Cyber Security, University of Electronic Science and Technology of China, Chengdu 611731, China; johnsonzxs@uestc.edu.cn

\* Correspondence: wangjf@scu.edu.cn; Tel.: +86-139-8096-7180

Received: 9 August 2017; Accepted: 19 September 2017; Published: 25 September 2017

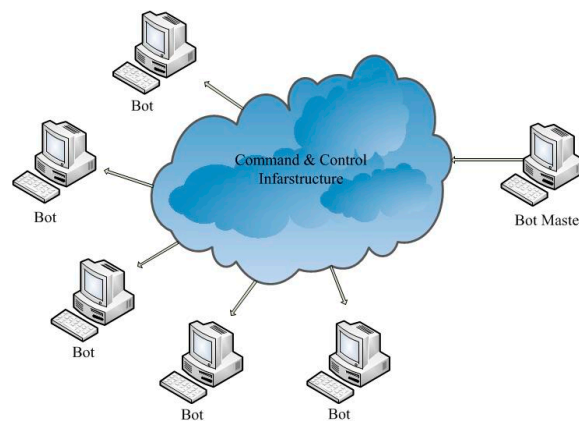
**Abstract:** With the help of botnets, intruders can implement a remote control on infected machines and perform various malicious actions. Domain Name System (DNS) is very famous for botnets to locate command and control (C and C) servers, which enormously strengthens a botnet's survivability to evade detection. This paper focuses on evasion and detection techniques of DNS-based botnets and gives a review of this field for a general summary of all these contributions. Some important topics, including technological background, evasion and detection, and alleviation of botnets, are discussed. We also point out the future research direction of detecting and mitigating DNS-based botnets. To the best of our knowledge, this topic gives a specialized and systematic study of the DNS-based botnet evading and detecting techniques in a new era and is useful for researchers in related fields.

**Keywords:** botnet; DNS; fast flux; domain flux; DGA; C and C

## 1. Introduction

Nowadays, botnets are considered as one of the most dangerous types of cyberattacks that may be controlled from long-range by their operators through C and C channels because they involve the use of very large coordinated groups of hosts simultaneously. Botnets are generally used for economic interest through launching cyberattacks, such as distributed denial of service (DDoS) [1–3], brute-force cracking, identity theft, adware installation, mass spamming, click frauds, and cyber warfare, etc. Botnets have been recently ported to the Internet of Things (IoT). A recent prominent example is the Mirai botnet (Japanese for “the future”) [4–6]. This malware compromised of IoT devices typically runs on multiple platforms, primarily spreads by first infecting devices, such as compromised refrigerators, security cameras, DVRs, and other consumer networking equipment. Therefore, it is without a doubt that botnets represent a major threat to the Internet infrastructure itself.

A botnet is a network of computers that have been linked together by malware, which are called bots, controlled by an unkind and cruel controller called the botmaster. With the constant sophistication and the resilience of them, botnets have been a serious threat to Internet security. Botnet activity can be divided mainly into three stages: spread, command and control, and launch of a cyberattack. The command and control is the core part of its working mechanism. The botmaster carries out a remote control on these bots and orders them to implement vicious activities through instructions. The method in which the bots are controlled depends upon the framework of the botnet command and control. Figure 1 illustrates the structure of a typical botnet. The foundation of any botnet is harmonious cooperation, which is how zombie machines identify and communicate with their controllers. Typically, zombie machines use their IP address, DNS name, or node id to locate the control servers.



**Figure 1.** Structure of a typical botnet.

We know that botmasters use various Internet protocols to conceal their malicious activities and avoid detection. In the past few years, some protocols have been heavily abused, and in recent years, DNS has become the main target of such malicious cyberattacks [7], such as Advanced Persistent Threat (APT) [8]. Malicious domains are basic tools in the hands of cybercriminals. Once a victim is bot-infected, the bot will tend to connect malicious domains to conduct actions over the Internet, such as awaiting the remote control command or delivering the bot reported feedback.

Experienced botnet developers have successfully set up a P2P botnet in some cases, which can completely avoid the use of a domain name, and most modern botnets frequently utilize the DNS system to support their C and C infrastructure [9]. In recent years, we have seen significant use of domain generation algorithms (DGAs) in the main botnet malwares [10] which have greatly enhanced the ability of botnets to evade detection or clearance. This is probably because it is very easy to develop and administrate the botnets built based on DNS. Compared with their P2P-based counterparts, this kind of botnet can offer excellent agility and increase detection difficulty. Moreover, some effectiveness of botnet mitigation measures in the past years has resulted in some collateral damage, triggering technical and policy-related problems.

The main purpose of this paper is to concentrate on botnet detecting techniques based on DNS and present a review of this field for a general summary of all these contributions. In this paper we have investigated the use of DNS in evasion or detection of botnets. The literature contains a significant amount of work regarding this branch of botnet detection, and we have collected and analyzed these works. To the best of our knowledge, this topic gives a specialized and systematic study of the DNS-based botnet detecting techniques in a new era. In this article, we research the application of DNS as the distribution channel of malicious payload. We represent the technical key problems and sum up some typical research findings of DNS-based botnet cyberattack detection. We believe that our work can help researchers and technicians to grasp the core issues and find future research works in this domain.

Our main contributions are as follows: We generalized the latest botnet flux characteristics and domain generation algorithms, as well as detecting technology for the study community. We illustrated how these attackers utilize these techniques to enhance their vitality in opposition to the detection. Corresponding with botnet evolution, we have investigated some recent studies that can explore or alleviate the attack of flux- and DGA-based botnets to a certain degree.

The remainder of this article is structured as follows: Section 2 discusses the evading techniques of botnets. Section 3 analyzes and summarizes the existing research works, regarding how to discover and detect DNS-based botnets. In this section, we analyze and compare some current detecting methods and strategies of DNS-based botnets. In Section 4, we propose the challenges and future works.

## 2. Botnet Evading Technology

A botnet is distinct from other malware, because the bot-herder can remote control and coordinate all the infected machines through a C and C server. In essence, this means that the control center of the botnet exists in the C and C server. This could be its Achilles heel. Hence, botnets put large amounts of resources into concealing the C and C server.

The DNS system is one of the most important elements of the Internet; it translates a domain name into an IP address, or vice versa. Quite notably, DNS helps Internet users locate various online resources, such as web servers and mail servers. Unfortunately, in view of its basic functions, the DNS service is frequently involved in various malicious activities one way or another. This situation is not surprising. Except for being used for well-known benign applications, domain names are widely used for malicious purposes. For instance, DNS are playing a more and more important role in the location of botnet C and C servers, scam servers, URLs of spam mails, malicious code downloading sites, and phishing pages from unsuspected victims. Hence, it seems useful to monitor the application of the DNS system because it indicates a certain domain serving as a part of malicious activities.

In order to avoid detection and mitigation [11], attackers have developed a series of more complicated technology to change C and C servers' IP addresses dynamically, such as DGA and domain fluxing.

### 2.1. Fast Flux

"Fast-flux" means that one can quickly assign different IP addresses to the same domain name. Generally speaking, this technique is used on legitimate occasions, but recently, it is frequently used for collaborative cyberattacks by cyber criminals. HoneyNet was the first one who observed this, and it was discovered that this technique had been utilized by some cyber criminals in many other illegal activities, such as phishing, spam, and malware spreading, etc., which are linked to criminal organizations. Fast-flux is able to make Internet cyberattacks more resistant to discovery and counter-measure if it is combined with peer-to-peer networking, proxy redirection, web-based load balancing, as well as distributed command and control.

Various different IP addresses can be mapped by one domain name very quickly in fast-flux. It is known that fast-flux can be used in a variety of legitimate activities; for instance, it can be used to improve the reliability of the service of highly-targetable networks. What is more, fast-flux can help organizations which provide content distribution service to balance their loads among a group of servers. Nevertheless, it is a double-edged sword, for it can also be employed by cyber criminals to launch collaborative cyberattacks.

Fast flux is a kind of dynamic proxy technology implemented by DNS. Its basic principle is that the use of some public IP address of the bot is used as the flux-agent, the C and C server domain name is resolved to the IP address of this flux-agent; the real server hidden behind these flux-agents provides services. To maintain the availability and concealment, the IP address of the flux-agent associated with a domain name are constantly changing. Using fast-flux technology, the botmaster can form a dynamic agent network with many bots, so it is difficult to find the hidden control server. In addition to the domain name of the botnet C and C server, the fast-flux network was also used to resolve the domain name of malware, phishing, and other malicious websites.

Botnets often employ fast-flux to control compromised hosts so as to turn them into a high-availability, load-balancing network that is similar to a content delivery network (CDN). It provides methods for load balance and redundant services. Botnets rapidly utilize content delivery networks to change IP addresses associated with both botnet infrastructure and spam websites. Such a phenomenon is called fast-flux, and the resultant structure is what we know as a fast-flux service network (FFSN), which has emerged in recent years, and is widely used in malicious network-type service. It adopts the fast flux technology to escape IP detection to ensure the service is online at all times.

Fast fluxing includes two types: single fluxing and double fluxing. Single fluxing can be regarded as a special case of fast fluxing, because it assigns multiple IP addresses to a domain name. It is

well known that such IP addresses are able to be registered or de-registered quickly. This is why it is called fast-flux. Within a very short TTL (time to live) time, this IPs are able to map to some designated domain names (e.g., DNS A, CNAME records) in a circular manner. Double fluxing is an upgraded version of the single-flux technique, which is capable of fluxing IP addresses of the associated fully-qualified domain names (FQDN) and IP addresses of the responsible DNS servers (NS records). Then, the DNS servers are employed to transform the FQDNs into their corresponding IP addresses. Hence, redundancy and a higher level of protection are ensured by this technique. A single-flux service network differs from a double-flux service network in the following ways:

If the bots are clear about the DNS name associated with the C and C server, communication with the domain names occurs in two steps: (1) Resolve the domain name to an IP address. The name server responsible for the requested domain will offer this information; (2) Send a request to the resolved IP address.

Single flux targets are involved in the second step of the above-mentioned method. Bots do not have direct communication with the C and C server. In fact, an intermediate layer of machines act as proxies and relay the communication between bots and the C and C server. Infected by the botnet, these machines are called 'proxy bots'. These resolved IP addresses represent the proxy bots. At a short time interval, a different subset of the entire pool of proxy bot IP addresses is correlated to the domain name. A collection of proxy bot IP addresses is employed by the botmaster in a circular manner; therefore, the same IP addresses may reappear in the DNS A record regularly. To order that all bots see such frequent change in the domain's record, the TTL of a name server's response is set to be very short, this is usually a few seconds. After being employed by botnets for a short time, a domain will be replaced by a new one, and the same set of proxy bots will be correlated with a new domain name by the botnets.

By extending the concept of flux to the name server responsible for resolving the C and C domain name, double fluxing brings the fast fluxing to a new level. The name server under the botnet's control (referred to in step (1)) will resolve a query request for the C and C domain name. The IP addresses correlated to the name server change rapidly. The name server's reply will contain the IP addresses correlated to the queried domain name. The resolved IP addresses change very rapidly and correspond to the proxy bots that transmit information to and from bots and the C and C servers (step (2)).

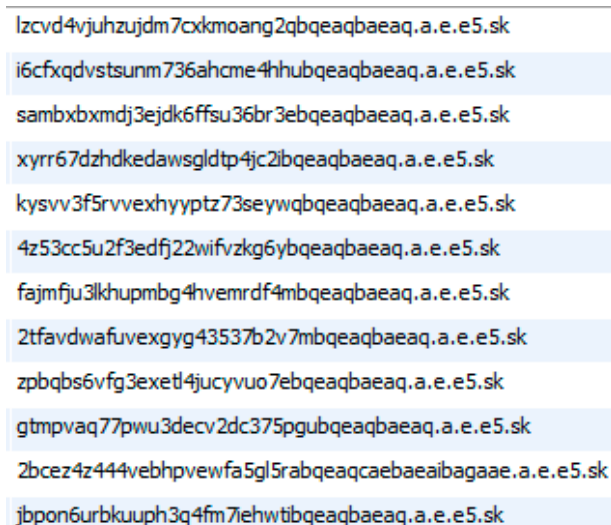
## 2.2. Domain Flux

Domain flux refers to a technology for keeping a malicious botnet in operation by constantly changing the domain name of the botnet owner's C and C server. When this technology is applied, a large number of candidate C and C server domain names may be generated by some algorithm, and the bots then try to connect to an active C and C server by querying the DNS server in turn. Using this technique, the botmaster can transfer C and C servers on multiple domain names flexibly. The past several years have witnessed extensive use of DGAs in such botnet crimewares as newGoZ, Torpig, and Conflicker. Owing to the application of DGA, every independent bot is able to generate numerous pseudo-random domains dynamically. Then an independent bot tries to connect the generated domains sequentially until the domain resolves to an IP address and the corresponding server generates an effective response or aborts after trying many times. If several generated domains are successfully registered by the botmaster, they can serve as C and C servers. For instance, every day, the Conflicker.C botnet algorithm is able to generate 50,000 different domain names in 110 top-level domains [12], among which, Conflicker.C attempts to connect to 500 domains; given the botmaster registers one domain every day there will be 1% of chance for the generated domains to get updated. This technique has a good ability to resist shut down. On the one hand, it has the large number of domain names; on the other hand, these domain names are across multiple management areas. It is difficult to register, block, or shut down these domain names. Even tracking the state of these domain names requires a large amount of various resources.

Generally speaking, the botmaster can transmit messages quickly to all bots, which is one of the major advantages of a botnet scheme. Meanwhile, there is a disadvantage, for the C and C server forms a single point of failure (SPOF). Therefore, the complete network will collapse if the server shuts down. To avoid the single point of failure limitation, one can adopt a C and C architecture with fault tolerance at times. This is true with the case of Conficker and, in this case, the domain name generation algorithm is applied to determine, in each case, the central server with which the bots are connected.

DGA is carried out to generate pseudorandom domain names by attackers dynamically [10]. DGA have many special features. Firstly, it is not necessary for the attackers to hard-code the C and C domains in the malware and the analysts can hardly decide which domain will be used for C and C purposes. Additionally, one can also use DGA for safe backup channels to back up situations when the initial communication channels fail; the bot can enter into a failsafe mode and begin to use the DGA for communication. For instance, DGA was used by Zeus GameOver as a safe backup communication mode [13]. In addition, DGA supports DNS fluxing (domain names which can be quickly registered and deregistered) in which the IP addresses of dynamically-generated multiple domains are frequently changed by attackers to evade the blacklisting of IP addresses.

We can carry out DGA in many ways. Firstly, by generating a hash and transforming it into an ASCII string appended with the top level domain (TLD), such as .com or .org, etc., we are able to generate pseudo-random domain names. In order to produce as many domains as possible, the logic should be deployed in a loop. Under such a situation, numerous DNS names can be produced by attackers, and then one name can be chosen and registered for C and C communication. For example, Figure 2 shows that we captured suspected domain names in our campus network.



```

lzcvd4vjuhzujdm7cxkmoang2qbqeaqbaeq.a.e.e5.sk
i6cfxqdvstsunm736ahcme4hhubqeaqbaeq.a.e.e5.sk
sambxbxmdj3ejdk6ffsu36br3ebqeaqbaeq.a.e.e5.sk
xyrr67dzhkedawsgldtp4jc2ibqeaqbaeq.a.e.e5.sk
kysvv3f5rvvexhyptz73seywqbqeaqbaeq.a.e.e5.sk
4z53cc5u2f3edfj22wifvzk6yqbqeaqbaeq.a.e.e5.sk
fajmfju3lkhupmbg4hvemrdf4mbqeaqbaeq.a.e.e5.sk
2tfavdwafuvexgyg43537b2v7mbqeaqbaeq.a.e.e5.sk
zpbqbs6vfg3exetl4jucyvuo7ebqeaqbaeq.a.e.e5.sk
gtmpvaq77pwu3decv2dc375pgubqeaqbaeq.a.e.e5.sk
2bcez4z444vebhpviewfa5gl5rabqeaqcaebaeibagaae.a.e.e5.sk
jbpon6urbkuuph3q4fm7iehwitibqeaqbaeq.a.e.e5.sk

```

Figure 2. Suspected domain names.

Attackers can use a similar technique to establish URLs, which point to infected websites, from which Internet users download malicious files through drive-by download attacks. DGA provides bot-herders an edge through making the C and C architecture more powerful and portable. Consequently, DGA can now be regarded as a feature of financial botnets based on HTTP. The majority of botnets based on HTTP [14,15], such as Shylock, Carberp, and Zeus, utilize DGA to empower their C and C communication channels; therefore, it is more complex to fingerprint the C and C servers. The DGA implementation in Zeus and Carberp botnets will provide attackers a safe fallback mechanism in case P2P communication fails. DGA is often chosen for underground communication by cybercriminals.

DGA can be used to remarkably empower a botnet's ability to evade detection or takedown. It is very difficult to preemptively eliminate C and C channels (e.g., blacklisting or pre-registration) owing



to the very large number of potential rendezvous points. Additionally, given the present C and C domains or IPs are detected and taken down, the bots will be able to identify the relocated C and C servers when looking up the next set of domains generated automatically. In addition, because the use of public-key encryption, it is impossible to mimic communication from the botmasters for the bots will not accept any instructions not signed by their botmasters.

### 3. Botnet Detection Technology

Now, we turn from the botnet evading techniques to focus on the botnet detecting techniques. The data source of botnet detection techniques is mainly the network traffic used, as well as some application data (such as mail records and DNS logging). This article is mainly based on DNS detection technology. Moreover, the definition of abnormal patterns are mainly based on the characteristics of content (signature-based) anomalies and based on specific behavior (behavior-based) anomalies.

Bots are often detected through their communication with a C and C infrastructure [16]. To evade detection, botmasters increasingly obfuscate the C and C communications, for example, by utilizing fast flux or P2P protocols. At present, a botmaster that gives out instructions often monitor and control bots. The transmission of these orders, which are known as C and C messages, can be centralized, peer-to-peer or hybrid. In the centralized structure, the bots contact with the C and C servers to receive instructions from the botmaster. In comparison with other architecture, the speed of message propagation and convergence is faster. It is not difficult to carry out, maintain, and monitor. However, a single point of failure limits it. While the access to the C and C server is blocked, such botnets can be taken down. In the course of communication, centralized infrastructure is often adopted by IRC or HTTP botnets.

In addition to the network communication, some of the botnet behavior characteristics, such as community similarity, also reflected in the use of some network services (such as DNS or email log records). The researchers attempted to find the characteristics of abnormal records to identify possible botnets from this kind of data.

#### 3.1. FF-Based Botnet Detection

Nowadays, fast-flux is attracting more and more attention, and scientists have made great efforts to detect the damage caused by fast-flux based cyberattacks. Phishing websites and malware often use fast-flux services as proxies so that Internet users will not have access to upstream servers that contain reliable content. This is a deceptive behavior creating replicas of existing Web pages or other cyber resources to cheat a user to submit private information, such as their financial status, or PIN code. To hide these replicas, phishers need advanced technology and a high maintenance cost is involved.

Fast-flux networks and DNS domain fluxing, as C and C mechanisms, are adopted to control bots, which can help botnets, such as Storm, Conficker, Kraken, and Torpig [17], to achieve this purpose. A fast-flux network acquires a large number of proxies which redirect users' requests to the phishing websites. It is very difficult to perform a follow-up of the communications because the proxies change very quickly and they can take advantage of the short DNS TTL for the flux domain. It is of great importance for a fast-flux network owner to obtain a large number of proxies and heterogeneous locations of the proxies. Bots in a botnet act the role of proxies very well in a fast-flux network. This is why botnets can perform a hiding mechanism. For instance, the Storm botnet is able to hide code updating.

The HoneyNet Project [18] presents the first case that studied fast-flux networks, and this project describes several examples of fast-flux service networks. On this project's website, it gives a general introduction of fast-flux service networks, and tries to explain how they work and how criminals utilize them. FluXOR [19] was the first published system which was used to detect fast-flux networks. How FluXOR detects fast-flux networks depends on analyzing the observable characteristics by ordinary users. The authors of this paper proposed that three types of characteristics—domain name, availability of the network, and heterogeneity of the agents—can be used to differentiate between

benign and malicious hostnames. Holz et al. gave a definition to a metric which can be used to differentiate fast-flux networks from the legitimate content distribution networks [20]. This metric takes advantage of two limitations of fast-flux networks: (i) the diverse range of IP addresses of a fast flux; and (ii) lacking of guaranteed uptime of the flux agent, so the controlled machines can collapse anytime.

Later, the technology of data mining of live traffic was utilized by Nazario and Holz to discover new fast-flux domains, and then they traced those botnets with proactive measurements [21]. Caglayan et al. proposed a behavioral analysis of fast-flux networks by using their database [22]. The results indicate that such networks share common life-cycle features and form clusters, which are based on the size, growth, and the type of vicious behavior. In addition, Caglayan et al. came up with a study of an empirical method for checking and classifying fast-flux service networks immediately [23]. A passive approach for detecting and tracking malicious flux service networks was raised by Perdisci et al. [24]. Their detection system is based on the passive analysis of recursive DNS traffic gathered from multiple large-sized networks. It is capable of detecting raw malicious flux service networks rather taking them from spam mails or domain blacklists. It is worth mentioning that Bilge et al. present EXPOSURE, a system that utilizes massive and passive DNS analysis techniques to examine aggressive activity domains which are involved in [25]. They depict various properties of DNS names and the methods that they are queried, extracting 15 features from the DNS traffic grouped into the sets below: time-based features, DNS answer-based features, TTL value-based features, and domain name-based features. Simultaneously, for detecting malware-related domain names, Antonakakis et al. put out one detection system called Kopis [26]. Kopis governs DNS traffic at the high-class levels of the DNS hierarchy followed by orders, and it is able to precisely search the malware domains by checking for global DNS query resolution patterns. Compared with the DNS reputation systems presented previously, like Notos and EXPOSURE [27], which rely on supervising traffic from local recursive DNS servers, a new advantageous point offered by Kopis, it introduced a new traffic characteristic specifically chosen to leverage the global visibility gained by monitoring network traffic at the higher DNS hierarchy. Unlike the operations before, Kopis could make DNS administrators examine malware domains independently (i.e., without the requirement of other networks' data), so such a step can be used to prevent improper use with the wasting of resources. Moreover, Kopis is able to check malware domains even if all IP reputation information is unavailable.

Hu et al. proposed a DNS with a lightweight exploring engine, called DIGGER [28]. Their work takes the global IP-usage models that are demonstrated by various forms of vicious and gentle domains into account, with an aim at single and double fast-flux domains. These conclusions provide other scholars with discernments into the present general view of fast-flux botnets and their range in implementation, and reveal the potential trends for botnet-based services. Gržinić et al. raised a method for fast flux detection called CROFlux, relying on the passive DNS replication approach [29]. The proposed model can obviously reduce false positive detection rates, and are able to supervise dubious domains used for fast flux from others.

These above-mentioned contributions demonstrate that attackers are using fast-flux technologies, and more studies ought to be conducted. Discoveries in fluxing right now, like the short TTLs, fast DNS A and NS record alters, massive IPs, and self-governing system numbers (ASNs) in individual DNS replies, are connected with production networks without breaking laws, which lets fast-flux cyberattacks become uncertain as they exist alongside legal flux applications. In a word, the solutions of reducing damages enables the division into policy-based or detection-based solutions. Fast-flux reductions based on policy depend on a progressive DNS infrastructure to protect from fast-flux assaults. It is a management-oriented method with vast manipulative cost and a long progress of development. In contrast, detection-based methods are based on the technologies to distinguish and check fast-flux assaults. Detection-based reduction would be possible attractive aims in anti-fast-flux. Thus, it is advised that fast-flux networks ought to be researched in depth, for they establish a method with high promise of detecting botnets.

### 3.2. DGA-Based Botnet Detection

In the many typical forms of botnets, DGA-based botnets that use a domain name generation algorithm to avert detection, are some of the most destructive and hard to detect. They rely on DGAs to establish the available order and control infrastructures. Provided with the general trend of this mechanism, recent works have focused on the analysis of DNS traffic to identify botnets relying on their DGAs. Various technologies have since been designed to detect DGA domains in DNS traffic, containing analyzing algorithmic models of domains, reverse-engineering malware instances [30,31], grouped into non-existent domains in DNS lookups [32,33], behavioral models [34,35], Social Network Analysis [36], power spectral density (PSD) analysis [37], and directly capturing C and C traffic [38,39]. However, influencing detected DGA domains to create a practical fix to botnet threats in large-scale networks is currently restricted.

Some previous studies depicted the application of DNS as a C and C channel containing the detection of vicious DNS traffic. Yadav et al. established an approach to identify domain fluxes in DNS traffic by searching for patterns which exist in domain names generated algorithmically [10]. The authors make use of statistical approaches to classify vicious domains generated algorithmically from DNS traffic. Massive botnets use DGAs to build complicated C and C infrastructures for DDoS attacks. Schiavoni et al. established an approach named Phoenix to identify the botnets relying on DGA utilizing IP-based traits [40], with the exception of distinguishing DGA- and non-DGA-generated domains utilizing a different combination of string and IP-based characteristics, and discovering representations of botnets from the clusters of DGA-generated domains. The Phoenix approach consisted of three modules: a detection module, a discovery module, and an intelligence and observation module. The discovery module identifies DGA-generated domains while the detection module monitors the domains whose names are generated automatically. The intelligence and observations module gathers the results of the first two modules and extracts the meaningful information from the observed data.

In [41], Sharifnya and Abadi presented a reputation system to detect DGA-based botnets. Their primary purpose is to give a high negative reputation score automatically to every host which is involved in suspicious botnet activities. In order to reach this goal, firstly, they choose DNS queries with similar features at the end of every time window. Secondly, they identify the hosts that algorithmically generated a large set of suspicious domains and then they put them into a so-called suspicious domains group activity matrix. They also identify hosts with high numbers of failed DNS queries and put them into a so-called suspicious failure matrix. Finally, they calculate the negative reputation score of each host in these two matrices and label hosts with high negative reputation scores as bot-infected.

Grill et al. propose a technique for detecting malware using DGAs. It is capable of evaluating data from large-scale networks and without reverse engineering a binary or performing non-existent domain (NXDomain) examination [42]. They use a statistical approach and model the ratio of DNS requests, visited IPs for each host in the local network, and label the deviations from this model as DGA-based malware. They expect the malware try to resolve more domains during a minor interval without a corresponding quantity of newly-visited IPs. For this purpose, the only thing they need is the NetFlow/IPFIX statistics collected from the network. Almost any modern router or three-layer switch can generate these. By using this approach, DGA-based malware could be identified with zero to very few false positives. Simple and effective, the approach can detect data from large-scale networks with the lowest computational cost.

In [43], Bottazzi et al. intended to show how heuristics applied to large-scale proxy logs, considering a typical phase in the life cycle of botnets, such as the search for C and C servers through algorithmically-generated domains, may provide effective and extremely rapid results. Mowbray et al. proposed a DGA detecting procedure from DNS query data [44]. The way it works is to identify the client IP address with an unusual distribution of second-level string length in the domain that they query.



Antonakakis et al. propose a technology for detecting randomly-generated irreversible domains [45]. Their theoretical basis is that most of the random DGA-generated domains that a bot queries would result in NXDomain responses and that bots from the same botnet (generated by the same DGA algorithm) would produce similar NXDomain traffic. They use a combination method of clustering and classification algorithms. The clustering algorithm groups domains based on the similarity in the structure of domain names as the same as the groups of machines that queried these domains. The classification algorithm is designed to assign the generated clusters to models of known DGAs. If a clusters cannot be assigned to a known model, this indicates a new DGA variant or family model is to be generated. Their work concentrated on detecting DGAs, one component of financial botnets.

Recently, Tzy-Shiah Wang et al. proposed a DGA-based botnet-detecting scheme. This scheme is designated as DBod, which relies on an analysis of the behavior of the query of the DNS traffic [32]. The scheme makes use of the fact that hosts infected by the same DGA-based malware query the same sets of domains in the domain list. Most of these queries fail because a very finite number of the domains are, in fact, associated with an active C and C server.

Although a surplus of present studies on identifying DGA-generated domains in DNS traffic, coping with such threats still depends on detecting the DNS behavior of every single device. However, in large-scale networks with complicated DNS infrastructure features, we still cannot find the method, or sufficient resources, to exhaustively investigate each part of the network, which is necessary to identify each device infected in a short time. Therefore, there is a great interest to evaluate the species distribution of DGA-based bots inside the networks and to prioritize the remediation efforts [46]. The authors presented a tool, namely BOTMETER, to evaluate the amounts of DGA-based bot species over large networks using only DNS traffic observable at upstream DNS servers.

#### 4. Concluding Remarks

To prevent botnets, we generally install antivirus software, update virus signature files, and upgrade patches to clear the infected botnets and improve the host security levels. While strengthening the firewall security strategy, DNS blocking, router control techniques, etc., are fundamental to curb botnet attacks effectively.

DNS is a core part of the Internet. Gaining the attention of attackers, it is vulnerable to attack. DNS blocking is one of the main protection and preventive measures to protect the network security, which can be given to pass through according to the network traffic situation, or rules limit, to ensure the network is not overloaded.

Botnets are the most common vehicle of cyber-criminal activity. Moreover, nowadays botnets have become evasive. Increasing prevalence of botnets forces discovering suitable solutions with the times, but existing defense mechanisms are hardly able to catch up with the speed of botnet techniques.

Generally speaking, it is difficult to make an impartial comparison amongst different botnet detection methods because the assessment environment for each case is different, such as the number of hosts, runtime, and malware used. In all kinds of detecting methods for the botnet, the most effective one is the C and C server domain name request monitoring and filtering technology. By means of routing and DNS blacklists, blocking malicious IPs and domain names is a simple and effective technique. The DNS resolver service is the first step to connect with the Internet, so with the assistance of DNS traffic to monitor the botnet, the detection rate is highest.

Blocking C and C server request traffic and closing the C and C server shall be pursued at the same time, to fundamentally curb the botnet expansion. Based on a high detection rate and very good cross-platform characteristics, DNS traffic monitoring of a botnet is the most effective botnet detection and filtering technology. We can discover botnet detection through DNS, the advantages of which are a small traffic flow, low cost, real-time operation, and has no effect on network performance and traffic.

Recently, use of fast flux or DGAs as a C and C message-change mechanism has increased and it is expected that this popularity would continue developing in the long-term. It advises that

DNS-based botnet detection and takedown studies should be studied in depth. Certainly, it still requires coordinated effort, including law enforcement, security operators, and domain registrars across countries, around the world.

How to detect the infected host or existing botnet accurately is a key problem to deal with botnet threats. In botnet detection research there are two elements: one is the data source; the second is the definition of abnormal patterns. In addition, the accuracy of the detection method, performance, and deployment is an important index in this aspect of research.

In this paper, we provide a review on the research and development with respect to DNS-based botnets. We clearly realize that this is the initial measure in this field. We analyzed and made conclusions of most of the present studies about DNS-based botnets, containing theory, checking measures, and mitigation strategies. It would be helpful to carry out further research based on this work.

For future study, we plan to deeply evaluate and compare part of the aforementioned DNS-based detection methods of botnet in terms of efficiency, accuracy, etc. Moreover, we also envision a new detection algorithm specifically for dealing with both malicious and legitimate domains to detect botnets in our campus network.

**Acknowledgments:** This work was supported by the Jiangsu Future Networks Innovation Institute Prospective Research Project on Future Networks (BY2013095-3-08), the National Key Research and Development Program (2016YFB0800600), the National Natural Science Foundation of China (91338107, 91438119, 91438120), and the Provincial Key Science and Technology Research and Development Program of Sichuan, China (2016ZR0087).

**Author Contributions:** Xingguo Li, Junfeng Wang, and Xiaosong Zhang have read, analyzed the articles and synthesized the data.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Alieyan, K.; Kadhum, M.M.; Anbar, M.; Rehman, S.U.; Alajmi, N.K.A. An overview of DDoS attacks based on DNS. In Proceedings of the International Conference on Information and Communication Technology Convergence (ICTC), Jeju Island, Korea, 19–21 October 2016; pp. 276–280.
2. Gross, G. Detecting and destroying botnets. *Netw. Secur.* **2016**, *2016*, 7–10. [[CrossRef](#)]
3. Anagnostopoulos, M.; Kambourakis, G.; Kopanos, P.; Louloudakis, G.; Gritzalis, S. DNS amplification attack revisited. *Comput. Secur.* **2013**, *39 Pt B*, 475–485. [[CrossRef](#)]
4. Kolias, C.; Kambourakis, G.; Stavrou, A.; Voas, J. DDoS in the IoT: Mirai and Other Botnets. *Computer* **2017**, *50*, 80–84. [[CrossRef](#)]
5. Gardner, M.T.; Beard, C.; Medhi, D. Using SEIRS Epidemic Models for IoT Botnets Attacks. In Proceedings of the 13th International Conference on Design of Reliable Communication Networks (DRCN 2017), Munich, Germany, 8–10 March 2017; pp. 1–8.
6. Jerkins, J.A. Motivating a market or regulatory solution to IoT insecurity with the Mirai botnet code. In Proceedings of the 2017 IEEE 7th Annual Computing and Communication Workshop and Conference (CCWC), Las Vegas, NV, USA, 9–11 January 2017; pp. 1–5.
7. Anagnostopoulos, M.; Kambourakis, G.; Gritzalis, S. New facets of mobile botnet: Architecture and evaluation. *Int. J. Inf. Secur.* **2016**, *15*, 455–473. [[CrossRef](#)]
8. Zhao, G.; Xu, K.; Xu, L.; Wu, B. Detecting APT Malware Infections Based on Malicious DNS and Traffic Analysis. *IEEE Access* **2015**, *3*, 1132–1142. [[CrossRef](#)]
9. Rossow, C.; Andriesse, D.; Werner, T.; Stone-Gross, B.; Plohmann, D.; Dietrich, C.J.; Bos, H. SoK: P2PWNEED—Modeling and Evaluating the Resilience of Peer-to-Peer Botnets. In Proceedings of the 2013 IEEE Symposium on Security and Privacy (SP), San Francisco, CA, USA, 19–22 May 2013; pp. 97–111.
10. Yadav, S.; Reddy, A.; Reddy, A.L. Detecting algorithmically generated malicious domain names. In Proceedings of the 2010 ACM SIGCOMM Conference on Internet Measurement, Melbourne, Australia, 1–3 November 2010; pp. 48–61.
11. Ji, Y.; He, Y.; Jiang, X.; Cao, J.; Li, Q. Combating the evasion mechanisms of social bots. *Comput. Secur.* **2016**, *58*, 230–249. [[CrossRef](#)]

12. Porras, P.; Saidi, H.; Yegneswaran, V. A foray into Conficker's logic and rendezvous points. In Proceedings of the 2nd USENIX Conference on Large-Scale Exploits and Emergent Threats: Botnets, Spyware, Worms, and More, Boston, MA, USA, 22–24 April 2009; p. 7.
13. FBI Cracks \$100m Financial-Crime Botnet. Available online: <http://itp.net/mobile/598440-fbi-cracks-100m-financial-crime-botnet> (accessed on 26 December 2016).
14. Sakib, M.N.; Huang, C. Using anomaly detection based techniques to detect HTTP-based botnet C&C traffic. In Proceedings of the IEEE International Conference on Communications (ICC), Kuala Lumpur, Malaysia, 23–27 May 2016; pp. 1–6.
15. Acarali, D.; Rajarajan, M.; Komninos, N.; Herwono, I. Survey of approaches and features for the identification of HTTP-based botnet traffic. *J. Netw. Comput. Appl.* **2016**, *76*, 1–15. [CrossRef]
16. Asha, S.; Harsha, T.; Soniya, B. Analysis on botnet detection techniques. In Proceedings of the International Conference on Research Advances in Integrated Navigation Systems (RAINS), Karnataka, India, 6–7 May 2016; pp. 1–4.
17. Stone-Gross, B.; Cova, M.; Gilbert, B.; Kemmerer, R.; Kruegel, C.; Vigna, G. Analysis of a botnet takeover. *IEEE Secur. Priv.* **2011**, *9*, 64–72. [CrossRef]
18. The HoneyNet Project. "Know Your Enemy: Fast-Flux Service Networks". Available online: <http://www.honeynet.org/papers/ff/> (accessed on 19 May 2016).
19. Passerini, E.; Paleari, R.; Martignoni, L.; Bruschi, D. FluXOR: Detecting and monitoring fast flux service networks. In Proceedings of the 5th International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment (DIMVA '08), Paris, France, 10–11 July 2008; pp. 186–206.
20. Holz, T.; Gorecki, C.; Freiling, F.; Rieck, K. Measuring and detecting fast-flux service networks. In Proceedings of the 15th Network and Distributed System Security Conference (NDSS'08), San Diego, CA, USA, 8–11 February 2008.
21. Nazario, J.; Holz, T. As the net churns: Fast-flux botnet observations. In Proceedings of the 3rd International Conference on Malicious and Unwanted Software (MALWARE'08), Alexandria, VA, USA, 7–8 October 2008; pp. 24–31.
22. Caglayan, A.; Toothaker, M.; Drapaeau, D.; Burke, D.; Eaton, G. Behavioral analysis of fast flux service networks. In Proceedings of the 5th Annual Workshop on Cyber Security and Information Intelligence Research (CSIIRW'09), Knoxville, TN, USA, 13–15 April 2009; pp. 1–4.
23. Caglayan, A.; Toothaker, M.; Drapeau, D.; Burke, D.; Eaton, G. Real-time detection of fast flux service networks. In Proceedings of the Cybersecurity Applications and Technology Conference for Homeland Security (CATCH'09), Washington, DC, USA, 3–4 March 2009; pp. 285–292.
24. Perdisci, R.; Corona, I.; Dagon, D.; Lee, W. Detecting malicious flux service networks through passive analysis of recursive DNS traces. In Proceedings of the Annual Computer Security Applications Conference (ACSAC'09), Honolulu, HI, USA, 7–11 December 2009; pp. 311–320.
25. Bilge, L.; Kirda, E.; Kruegel, C.; Balduzzi, M. EXPOSURE: Finding malicious domains using passive DNS analysis. In Proceedings of the 18th Annual Network and Distributed System Security Symposium (NDSS), San Diego, CA, USA, 6–9 February 2011.
26. Antonakakis, M.; Perdisci, R.; Lee, W.; Vasiloglou, N., II; Dagon, D. Detecting malware domains at the upper DNS hierarchy. In Proceedings of the USENIX Security Symposium, San Francisco, CA, USA, 8–12 August 2011; p. 16.
27. Antonakakis, M.; Perdisci, R.; Dagon, D.; Lee, W.; Feamster, N. Building a Dynamic Reputation System for DNS. In Proceedings of the USENIX Security Symposium, Washington, DC, USA, 11–13 August 2010; pp. 273–290.
28. Hu, X.; Knysz, A.; Shin, K.G. Measurement and Analysis of Global IP-Usage Patterns of Fast-Flux Botnets. In Proceedings of the IEEE INFOCOM, Shanghai, China, 10–15 April 2011.
29. Gržinić, T.; Perhoć, D.; Marić, M.; Vlašić, F.; Kulcsar, T. CROFlux—Passive DNS Method for Detecting Fast-Flux Domains. In Proceedings of the 2014 37th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), Opatija, Croatia, 26–30 May 2014.
30. Barabosch, T.; Wichmann, A.; Leder, F.; Gerhards-Padilla, E. Automatic Extraction of Domain Name Generation Algorithms from Current Malware. NIAS, 2012. Available online: [http://four.cs.uni-bonn.de/fileadmin/user\\_upload/wichmann/Extraction\\_DNGA\\_Malware.pdf](http://four.cs.uni-bonn.de/fileadmin/user_upload/wichmann/Extraction_DNGA_Malware.pdf) (accessed on 19 May 2016).

31. Domain Name Generator for Murofet. Available online: <http://blog.threatexpert.com/> (accessed on 29 October 2016).
32. Thomas, M.; Mohaisen, A. Kindred domains: Detecting and clustering botnet domains using DNS traffic. In Proceedings of the International Conference on World Wide Web Companion, Seoul, Korea, 7–11 April 2014; pp. 707–712.
33. Heuer, T.; Schiering, I.; Klawnn, F.; Gabel, A.; Seeger, M. Recognizing Time-Efficiently Local Botnet Infections—A Case Study. In Proceedings of the 11th International Conference on Availability, Reliability and Security (ARES), Salzburg, Austria, 31 August–2 September 2016; pp. 304–311.
34. Erquiaga, M.J.; Catania, C.; García, S. Detecting DGA malware traffic through behavioral models. In Proceedings of the 2016 IEEE Biennial Congress of Argentina (ARGENCON), Buenos Aires, Argentina, 15–17 June 2016; pp. 1–6.
35. Wang, T.; Lin, H.; Cheng, W.; Chen, C. DBod: Clustering and detecting DGA-based botnets using DNS traffic analysis. *Comput. Secur.* **2017**, *64*, 1–15. [[CrossRef](#)]
36. Wang, T.; Lin, C.; Lin, H. DGA Botnet Detection Utilizing Social Network Analysis. In Proceedings of the 2016 International Symposium on Computer, Consumer and Control (IS3C), Xi'an, China, 4–6 July 2016; pp. 333–336.
37. Kwon, J.; Lee, J.; Lee, H.; Perrig, A. PsyBoG: A scalable botnet detection method for large-scale DNS traffic. *Comput. Netw.* **2016**, *97*, 48–73. [[CrossRef](#)]
38. Nadji, Y.; Antonakakis, M.; Perdisci, R.; Dagon, D.; Lee, W. Beheading Hydras: Performing Effective Botnet Takedowns. In Proceedings of the 2013 ACM SIGSAC Conference on Computer & Communications Security (CCS), Berlin, Germany, 4–8 November 2013.
39. Nelms, T.; Perdisci, R.; Ahamad, M. Execscent: Mining for new C&C domains in live networks with adaptive control protocol templates. In Proceedings of the USENIX Security Symposium, Washington, DC, USA, 14–16 August 2013; p. 589.
40. Schiavoni, S.; Maggi, F.; Cavallaro, L.; Zanero, S. Phoenix: DGA-based botnet tracking and intelligence. In *Detection of Intrusions and Malware, and Vulnerability Assessment, 2014*; Springer: Cham, Switzerland, 2014; pp. 192–211.
41. Sharifnya, R.; Abadi, M. A Novel Reputation System to Detect DGA-Based Botnets. In Proceedings of the International Conference on Computer and Knowledge Engineering (ICCKE), Mashhad, Iran, 31 October–1 November 2013; pp. 417–423.
42. Grill, M.; Nikolaev, I.; Valeros, V.; Rehak, M. Detecting DGA malware using NetFlow. In Proceedings of the IFIP/IEEE International Symposium on Integrated Network Management (IM), Ottawa, ON, Canada, 11–15 May 2015; pp. 1304–1309.
43. Bottazzi, G.; Italiano, G.F. Fast Mining of Large-Scale Logs for Botnet Detection: A Field Study. In Proceedings of the IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing, Liverpool, UK, 26–28 October 2015; pp. 1989–1996.
44. Mowbray, M.; Hagen, J. Finding Domain-Generation Algorithms by Looking at Length Distributions. In Proceedings of the IEEE International Symposium on Software Reliability Engineering Workshops, Naples, Italy, 3–6 November 2014; pp. 395–400.
45. Antonakakis, M.; Perdisci, R.; Nadji, Y.; Vasiloglou, N.; Abu-Nimeh, S.; Lee, W.; Dagon, D. From throw-away traffic to bots: Detecting the rise of DGA-based malware. In Proceedings of the USENIX Conference on Security Symposium, Bellevue, WA, USA, 8–10 August 2012; p. 24.
46. Wang, T.; Hu, X.; Jang, J.; Ji, S.; Stoecklin, M.; Taylor, T. BotMeter: Charting DGA-Botnet Landscapes in Large Networks. In Proceedings of the IEEE 36th International Conference on Distributed Computing Systems (ICDCS), Nara, Japan, 27–30 June 2016; pp. 334–343.

