Charlatans' Web: Analysis and Application of Global IP-Usage Patterns of Fast-Flux Botnets

Matthew Knysz, Xin Hu, and Kang G. Shin
University of Michigan Ann Arbor

Abstract

Botnet-based hosting or redirection/proxy services provide botmasters with an ideal platform for hosting malicious and illegal contents while affording them a high level of misdirection and protection. Because of the unreliable connectivity of the constituent bots (e.g., compromised home computers), domains built atop botnets require frequent updates to their DNS records, replacing the IPs of offline bots with active ones to prevent a disruption in service. Consequently, their DNS records contain a large number of constantly-changing (i.e., "fluxy") IPs, earning them the descriptive moniker of fast-flux domains—or, when both the content and name servers are fluxy, double fast-flux domains. In this paper, we examine the global IP-usage patterns exhibited by different types of malicious and benign domains, including single and double fast-flux domains. We have deployed a lightweight DNS probing engine, called *DIGGER*, on 240 PlanetLab nodes spanning 4 continents. Collecting DNS data for over 3.5 months on a plethora of domains, our global vantage point enabled us to identify distinguishing behavioral features between them based on their DNS-query results. We have quantified these features and demonstrated their effectiveness for detection by building a proof-of-concept, multi-leveled SVM classifier capable of discriminating between five different types of domains with minimal false positives. We uncovered new, cautious IP-management strategies currently employed by criminals to evade detection. Our results provide insight into the current global state of fast-flux botnets, including the increased presence of double fast-flux domains and their range in implementation. In addition, we expose potential trends for botnet-based services, uncovering previously-unseen domains whose name servers *alone* demonstrate fast-flux behavior.

I. Introduction

A botnet is a vast collection of compromised computers under the control of a botmaster utilizing a common Command-and-Control (C&C) infrastructure. By exploiting Internet Relay Chat (IRC), peer-to-peer (P2P), and other protocols as flexible and extensible means for C&C, botnets have gained a great deal of versatility in providing malicious services and generating profit. The ability to coordinate thousands of individual bots allows the botmaster to launch larger-scale, sophisticated attacks. Among the numerous criminal uses of botnets, one of the more advantageous is the botnet-based hosting service, which proxies or redirects unsuspecting users to illegal or nefarious content. Since botnets are essentially an abundant source of disposable IPs, they can easily be turned into a large network of front-end redirection/proxy servers pointing to malicious content hosted elsewhere—on anything from a powerful central server to another bot.

Used as a misdirection mechanism for evading detection, botnet-based hosting services often come in tandem with a variety of other criminal scams, constituting an essential portion of botnets' overall operation. For example, spam/phishing campaigns often utilize botnets for misdirection. They begin by using some spamming mechanism (e.g., a hijacked mail server and/or a botnet) to send seemingly interesting phishing emails. Within the phishing emails are innocuously disguised embedded links whose domain names resolve to IP addresses of compromised computers in a botnet. Once victims click the embedded links, they connect to the bots, which then redirect them to—or serve as proxies for—the central host (often called the *mothership*) of the nefarious content. This strategy grants criminals a high level of anonymity while enabling easy and centralized management of the malicious content. However, because botnets are composed primarily of compromised home computers with unreliable connectivity, it is not uncommon for them to unexpectedly go offline (e.g., the computer is turned off or the installed malware is discovered and removed). Botnet-based hosting services, therefore, must be protected against the failure or disruption of individual bots, ensuring the availability and stability of the hosted service/content. As a result, it is beneficial for bot-based hosting infrastructures to adopt fast-flux DNS techniques, which frequently change the domain name mappings to different bots' IP addresses. When the victim tries to visit the malicious domain, the DNS server responds with a set of up-to-date, active bot IPs. By recruiting a large pool of IPs and supplying a

large set of IPs per query, botmasters can ensure, with high probability, that the malicious domain resolves to at least one valid IP belonging to an online bot. An additional level of control and resilience is attained by giving the domain's IP mappings a short time-to-live (TTL) value. This permits botmasters a quick response when a bot goes offline, replacing its IP with one from the ample supply of online bots. Using this fast-flux technique, botmasters effectively turned their botnets into a global Content Delivery Network (CDN), providing highly available and reliable content-hosting services in spite of node failures. This extends the lifetime of illegal activities the botnets provide, complicating disruption efforts by introducing an additional layer of misdirection.

Previous research has studied the features of fast-flux botnets and their malicious uses in phishing scams [17] (e.g., Storm Worm and Rock Phish). However, little has been reported on botnets' IP-usage behavior from a global perspective. Because botnets are formed with myriad compromised hosts dispersed around the world, accurate characterization of how botmasters manage this vast number of IPs can only be achieved by collecting and analyzing data from a global viewpoint. In this paper, we attempt to fill this important gap and explore the global usage patterns of botnets' IP addresses. Our work is unique and different from the previous work in the following four ways. First, we build a global query engine called *DIGGER* that monitors complete DNS behavior from 240 geographically-dispersed vantage points for an extended period of time. This provides us with a unique, global view of how different types of domains differ in their IP-usage patterns. Second, we propose effective methods to characterize and quantify the temporal and spatial IP-usage patterns of fast-flux botnet domains, facilitating the classification and detection of different domain types. This also allows us to reveal several previously-unknown features of fast-flux botnets and uncover new, discreet IP-management strategies currently employed by criminals to evade detection. Third, we design and implement a proof-of-concept classifier based on a multi-leveled machine learning algorithm. Utilizing the behavioral features of a domain's IP usage, the classifier accurately and automatically identifies different types of malicious and benign domains. Finally, we apply the classifier on three months' worth of globally-collected data. The results demonstrate the current trend of fast-flux botnets and the effectiveness of the distinguishing behavioral features thanks to our global DNS monitoring system.

The remainder of this paper is organized as follows. Section II reviews related work. Section III defines the terminology we use. Section IV explores the global DNS IP-usage patterns for different domain types. Section V presents our proof-of-concept classifier and its experimental results, and finally, Section VI concludes the paper.

II. Related Work

Botnets have now become one of the biggest threats to Internet services and applications. Most previous research focused on understanding of the operations and threats of botnets by collecting and analyzing bot-related activities, such as IRC traffic [19], spam emails [26], DNS queries [20], and DNS Blacklist queries [21]. Rajab *et al.* [19] constructed a distributed infrastructure to measure the Internet Relay Chat (IRC) botnet activities and showed that botnets contribute the majority of unwanted traffic in the Internet. Collins *et al.* [6] explored the spatial and temporal uncleanliness of networks and showed the correlation between botnets and spam/scanning activities. Recently, botnets have appeared in the wild using P2P infrastructures for the C&C channel, making them more robust to node failures and difficult to take down. Grizzard *et al.* [8] analyzed the architecture and communication protocol of a most recent P2P botnet, Peacom (a.k.a. *Storm Worm*) [5]. A model for advanced hybrid P2P botnets has also been proposed in [24], which provides robust connectivity, control traffic dispersion, encryption, easy recovery and many other features. Most of these methods fall into the category of passive analysis. To gain an insider view of a botnet, researchers also took more active approaches, infiltrating botnets with actual malware samples or customized crawlers. For example, Holz *et al.* [14] crafted a specific P2P client to join the Storm Worm's P2P botnet and estimate the total number of compromised machines. Researchers also disrupted the Conficker botnet by sinkholing future DNS domains of the C&C server, preventing botmasters from updating the infected hosts [15]. More recently, Stone-Gross *et al.* [23] successfully took over the Torpig botnet for ten days by preemptively registering DNS domains the bots would be contacting as C&C servers in the near future. This allowed them to reveal detailed operations of the Torpig botnet and accurately estimate the number of compromised hosts.

Because of the significant threats botnets have posed on the Internet security, numerous detection approaches have been proposed based on the network or host behavior typical of bots. Rishi [7] passively monitors IRC traffic for suspicious IRC nicknames, IRC servers and uncommon server ports to detect bot-infected machines. Binkley and Singh [4] proposed detection of IRC-based botnets via TCP anomaly detection and IRC message statistics.

BotHunter [10] attempts detection using IDS-driven dialog correlation based on IRC C&C communication and other common actions taken during the life cycle of a bot. Meanwhile, to track and analyze botnets in a large tier-1 ISP, Karasaridis *et al.* [16] proposed a wide-scale detection technique that looks for typical network-flow patterns between bots and their controllers. BotSniffer [11] identifies HTTP- and IRC-based C&C channels by capturing the coordinated and synchronized communication patterns in the C&C traffic. To eliminate the reliance on IRC- or HTTP-based C&C protocols for identifying botnets, Gu *et al.* proposed BotMiner [9], which clusters similar communication and malicious traffic and performs cross-cluster correlation to identify potential bot-infected hosts.

Among the numerous criminal uses of botnets, their use as hosting or redirection/proxy servers for illegal content and phishing scams provides an ideal platform for financial gain. However, because of the unreliable nature of the bots, more and more botmasters have adopted fast-flux DNS techniques to ensure the availability and stability of their malicious service/content. Fast-flux techniques are characterized by the frequent change of domain name mappings to the IP addresses of different bots. Holz *et al.* [13] studied the characteristics of fast-flux networks and first developed detection algorithms; they extract URL links from spam emails and then identify fast-flux networks based on the number of unique IP addresses in DNS queries and the number unique AS to which the IPs belong. Nazario and Holz [17] applied a similar approach to track the use of fast-flux domains and characterize several features of fast-flux botnets, such as member size, lifetime, and top-level domain distribution. Their work demonstrated that continuous data mining of fast-flux DNS records can yield insights into the operations of fast-flux botnets. Despite the increasing awareness of fast-flux botnets to the security community [22], there has been little effort in understanding botnets' global IP-usage patterns of different types of fast-flux botnets (in particular, double fast-flux domains). We attempt to fill this important gap by continually monitoring the DNS properties of fast-flux domains from a large number of geographically-dispersed vantage points, allowing us to study their behavior patterns from a global perspective. In addition, since the purpose of using fast-flux botnets is to reliably distribute the illegal content to users despite host failures, the behavior of fast-flux botnets resembles that of traditional CDNs [3] like Akamai and CDNetworks. As a result, we conduct in-depth, comparative analysis of IP management of fast-flux botnets and popular CDNs. With this knowledge, we are able to develop algorithms that can accurately distinguish between the different types of fast-flux domains and discern them from other domain types—both benign and malicious.

III. Terminology

This section defines the terminology we have adopted for succinctness and clarification when discussing the various domain types and DNS records in this paper.

1) DNS records: We have gathered detailed logs of the DNS behavior for a large number of different domains (both valid and malicious) to enhance our understanding. Our data-acquisition process will be detailed in Section IV. For now, we need only explain a few terms used to describe particular components of a domain's DNS-query results.

A rec: is the A (address) record returned in the result of a DNS query on a *domain*. It contains the host IP addresses associated with the domain at that DNS server. An A rec, therefore, consists of the IP addresses of the actual machines hosting the domain's content.

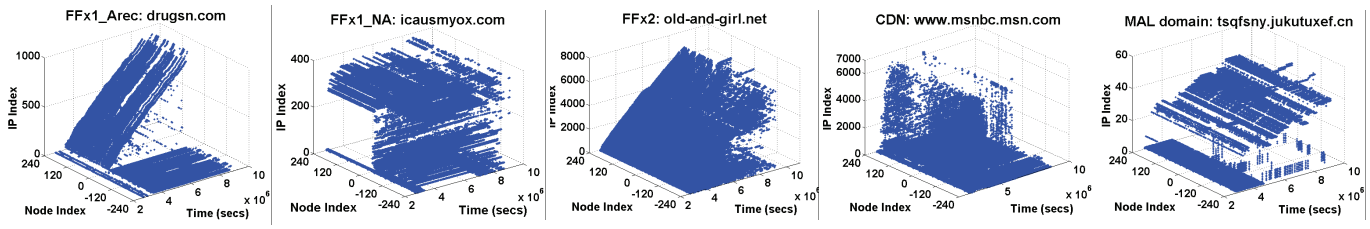
NS rec: is the NS (name server) record returned in the result of a DNS query on a domain. It contains the name servers (NSes) in charge of that domain. These records contain only the NSes' domain names and not their IP addresses. For example, an NS rec for the domain *www.example.com* may contain the NSes *ns1.example.com* and *ns2.example.com*, but not their IP addresses.

NA rec: refers to the A record returned in the result of a DNS query on a *name server*. An NA rec, therefore, consists of the IP addresses of the actual machines serving as the domain's name servers.

Reverse DNS lookup/name: is the result of a DNS-query request for the domain name of an IP address. When we perform a DNS query on a domain, we also perform a reverse DNS lookup on the IPs returned in the domain's A and NA recs. Because the reverse DNS names are set by the IP's Internet Service Provider (ISP) and not the domain's owner, they can be different from the original domain or NS domain names queried.

2) Domain types: We gathered extensive DNS-query results for a variety of domain types, including valid and benign domains as well as malicious botnet domains. In Fig. 1, we have plotted the global IP usage—as seen from the DNS queries—for some representative domains of the different domain types. In this figure, the *Time* axis represents the time (in seconds) since we started monitoring the domains; *Node Index* represents the node (from those dispersed around the globe) that the IP was observed on, with positive values indicating an A rec IP and

negative values an NA rec IP of the domains; *IP Index* is a unique index assigned, in ascending order, to each newly-observed IP. The following is an explanation of the terms we use to describe these various domain types and how they behave. Their global behavior will be explained further in Section IV.



1: Global IP usage (in DNS results) for some examples of the domain types

FF domain: is a malicious domain utilizing a Fast-Flux (FF) DNS-advertisement strategy. These domains are typically built atop botnets, since bots function as a readily available and disposable source of IPs for advertising to DNS servers. Because bots unexpectedly go offline, FF domains advertise numerous IPs in their DNS-query results, helping ensure some of the IPs belong to a functional bot. The TTL of the IPs used by FF domains tend to be relatively short; this permits the botmasters a finer level of control in replacing IPs advertised to the DNS servers, increasing the availability of an online bot and access to the malicious payload. It is this excessive number of constantly-changing IP addresses that qualifies a domain's DNS records and advertisement strategy as "fluxy", and the domain is considered a FF domain.

FFx1_Arec domain: represents a FF domain that demonstrates a FF DNS-advertisement strategy in its *A rec* DNS-query results, but not its NA rec. It is considered a single fast-flux domain (FFx1), since only its content servers (i.e., the A rec IPs) resemble a FF domain.

FFx1_NArec domain: represents a FF domain that demonstrates a FF DNS-advertisement strategy in its *NA rec* DNS-query results, but not its A rec. This is in direct contrast to its FFx1_Arec counterpart.

FFx2 domain: (double fast flux) is, as its name suggests, the composite of FFx1_Arec and FFx1_NArec domains. The FF DNS-advertisement strategy described previously can clearly be witnessed in both its A and NA recs, implying the use of bots for its content and name servers. A FFx2 domain can provide unprecedented control in the management of the domain and its resources—botnet or otherwise—with the DNS service, affording the botmaster a high level of misdirection and protection.

FFx1 domain: (single fast flux) is a domain exhibiting FF behavior in its A or NA record, but not both.

CDN domain: is a valid, benign domain that uses a CDN, such as Akamai, to improve the delivery of its content. CDNs—consisting of a system of computers networked together for the purposes of improving the performance and scalability of content distribution—produce DNS-query results resembling those of malicious FF domains: numerous IPs per query with short TTL values. This affinity is a consequence of their similar goal to provide reliable content delivery despite node failure, as well as their shared assumption that any node can temporarily or permanently fail at any time. However, CDN domains tend to demonstrate geographic awareness (i.e., IPs that are geographically close to a DNS server will be advertised with higher probability at that server) and load balancing—advanced techniques for improving performance and scalability not yet observed in FF domains.

Non-CDN domain: is a valid, benign domain that *doesn't* use a CDN for delivery of its content. Typically, a non-CDN domain uses a few stable content servers and a modest number of NSes; the same IPs of the content and name servers appear in DNS-query results regardless of the geographic location of the DNS server queried.

MAL domain: is a malicious domain that isn't fluxy enough to be considered a FF domain. However, it also isn't benign enough to be considered a non-CDN domain. It tends to recruit more IPs than a non-CDN, but not nearly as many as a FF domain. For example, during a monitoring period of a few months, a FF domain is likely to advertise thousands of different IPs with DNS; even a fairly slow FF domain will advertise in the hundreds. A MAL domain, on the other hand, will advertise perhaps a total of 20-30 IPs—roughly one or two IPs every few days. This is different from a non-CDN. While a non-CDN may have 20-30 IPs, they are all seen essentially at once and are stable for the duration of the monitored period. A MAL domain may have some stable IPs over the monitored period, or they may not, and the IPs will eventually be replaced by new ones. A MAL domain will tend to slowly add more IPs because they will slowly lose some as their malicious activities are detected and their IPs

are blocked. The IPs used by a MAL domain may consist of bots or valid servers being used for malicious means. Unlike valid domains, MAL domains will exhibit some IP overlap (i.e., the same IPs appear in both the A and NA recs). If a MAL domain is using bots, a reverse DNS lookup can reveal the presence of compromised home computers, although this isn't always the case; often, MAL domains exclusively use stable servers from hosting providers until the IP is blocked.

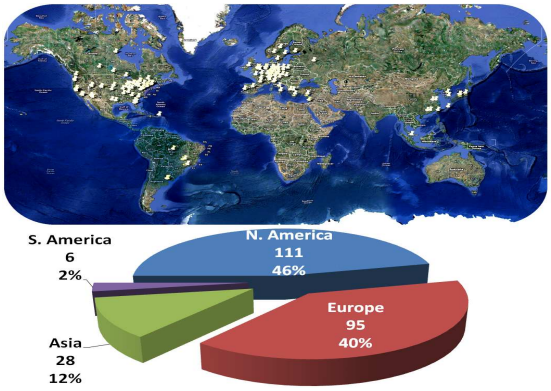
IV. Global IP Usage Patterns of Botnet

A. Overview

In this section, we explore the DNS IP-usage patterns of the previously-described domain types, identifying interesting and differentiating features among them. We accomplish this by analyzing numerous domains' DNS-query results from vantage points dispersed around the world. This provides us with a unique, *global* perspective of how the different types of domains advertise their IP addresses to DNS servers. First, we will describe how we set up a globally-distributed DNS monitoring system and then discuss the various features we have identified that could be useful in the detection and classification of CDN, non-CDN, and the different FF domains. Lastly, we will show how some of these features differ for MAL domains and how these variances could aid in their classification.

B. System Architecture

To understand how the IP-usage patterns for FF botnet domains differ from valid (e.g., CDN) domains on a global scale, we created a distributed DNS-query engine called DIGGER, deployed on 240 geographically disparate nodes in the PlanetLab testbed [18]. The nodes were chosen based on the location of the DNS servers they queried, such that DIGGER would issue queries to DNS servers in different geographic locations around the world. Fig. 2 shows the distribution of DIGGER nodes, which is reflective of the overall distribution of available PlanetLab nodes.



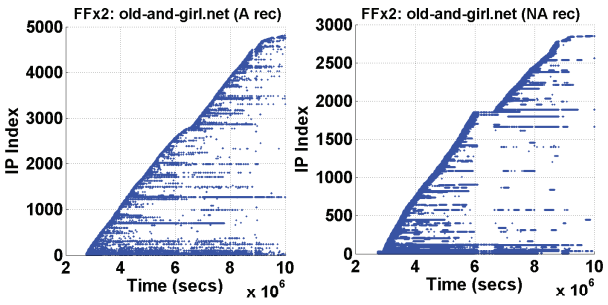
2: Global distribution of DIGGER nodes by continent

Domain Type	Domain	A rec	NA rec	Overlap
FFx1_Arec	drugsn.com	932	33	0
	www.couldchoose.com	486	37	5
FFx1_Narec	icausmyox.com	16	370	1
FFx2	old-and-girl.com	5,227	3,047	879
	mountainready.com	4,060	2,219	2,144
MAL	duelready.com	16	32	15
	tsqfsny.jukutuxef.cn	23	42	20
CDN	www.msnbc.msn.com	1,160	5,412	0
non-CDN	hostingprod.com	18	32	0

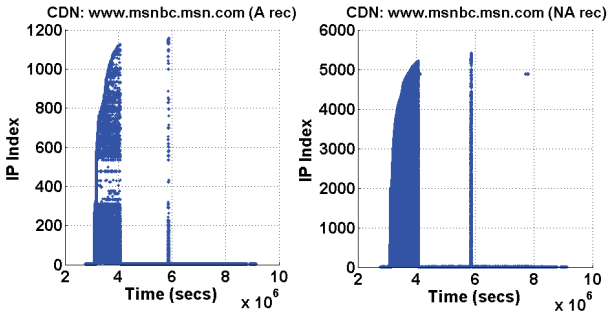
I: Total A and NA rec IPs and IP overlap for different domain types.

On each node, DIGGER performs intelligent DNS query digs on a set of malicious and benign domains, monitoring the returned results. For each domain, DIGGER digs the domain's A rec, NS, NA rec and the reverse DNS lookup for the A and NA rec IPs. From this data, we can determine the IPs being used to serve the domain's content as well as the IPs being used for the NSes. With the reverse DNS lookup, we can more easily identify IPs belonging to compromised computers. Since compromised home computers constitute a large portion of botnets, any reverse DNS lookups resulting in names typical to home computers (i.e., containing words like comcast, charter, broadband, dailup, etc.) are highly indicative of a potential bot.

Based on a domain's most recently returned DNS-query results, DIGGER classifies the domain as either active or offline. DIGGER continues to dig active domains periodically based on their observed maximum TTL, eliminating wasteful DNS queries while ensuring fresh DNS-query results. Domains that have been determined to be offline are intermittently dug, so that DIGGER can determine if they come back online later. Every 24 hours, DIGGER compresses the raw DNS-query data and uploads the results to our centralized server for analysis. As our central server gathers the compressed DNS-query results from DIGGER, it automatically parses them into a more compressed and useful format for feature extraction, removing any invalid queries. This way we aggregate



3: IP usage for old-and-girl.net (FFx2)



4: IP usage for www.msnbc.msn.com (CDN)

the global DNS-query results for over 106,000 different domains from 240 nodes around the globe. The set of domains monitored by DIGGER was compiled from multiple sources, including online repositories of phishing [2] and malware [1] websites. In addition, we extracted domains from URL links embedded in spam emails found in our personal mail boxes as well as online repositories [12].

DIGGER has been deployed and gathering global DNS-usage patterns for a little over 3.5 months. Based on the analysis of this data, we have identified several differentiating features between malicious FF botnet domains and valid domains, as described in the following subsections.

C. Overlap between IPs of A and NA Records

While analyzing our data, it quickly became apparent that FF domains tend to exhibit some IP overlap. We were seeing IPs advertised for a domain's A rec reappearing in the same domain's NA rec. Furthermore, when DIGGER would perform a DNS dig on the domain's NSes, the same IP would often be returned for different NSes. It became apparent that the malicious domains were not only reusing their available IP pool for both A and NA recs, but were also returning IPs from the same IP pool regardless of which NS was queried.

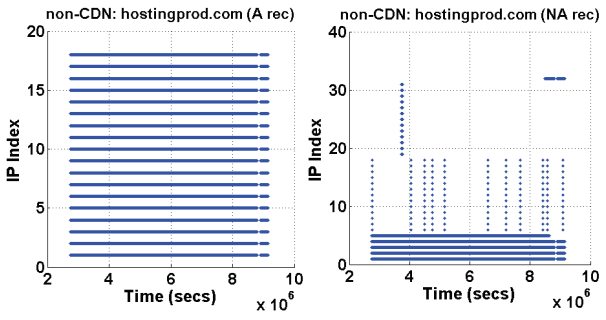
Table I shows the total number of A rec, NA rec, and overlap IPs (i.e., IPs appearing in both the A and NA rec) for some representative domains from each domain type. This overlap phenomenon was, as expected, much more prevalent in FFx2 domains than either type of FFx1; we never observed it in valid domains. The FFx1 domains almost entirely use valid IPs for one record type and the IPs of compromised computers for the other.

The IP overlap we have empirically observed is in line with our expectation. For redundancy and fault-tolerance purposes, a valid domain should almost always have separate machines serving as content providers and NSes. Otherwise, the domain may easily suffer from a single point of failure. A FF domain, on the other hand, will attempt to make the best use of all its limited resources, and thus, it will tend to reuse the IPs of compromised computers for both its A and NA records. Clearly, the amount of observed IP overlap could prove a useful feature for differentiating between valid and malicious domains, especially FFx2 domains.

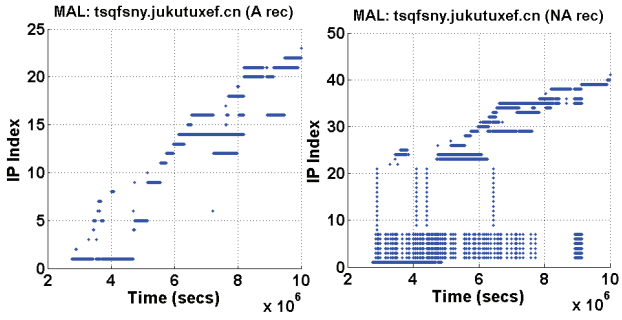
D. IP Recruiting

Due to their different resources and management techniques, one would expect FF, CDN, and non-CDN domains to demonstrate distinct strategies when advertising IPs to DNS servers. To test the validity of this expectation, we have analyzed the advertisement strategies for the various domain types. For a given domain, we assumed a global vantage point and assigned a unique IP index (in ascending order) to each newly seen IP in the DNS query results. This IP index is plotted against time for a FFx2 domain, a CDN domain, and a non-CDN domain in Figs. 3, 4, and 5, with the y-axis representing the unique IP index and the x-axis representing the time in seconds since DIGGER started monitoring domains. The points in the graphs represent when an IP was returned in a DNS query on a global scale (i.e., across all nodes monitored by DIGGER). Thus, the slope of each plotted curve demonstrates the rate, or speed, with which a domain—from the global vantage point —seems to "recruit" more unique IPs.

It should be noted that, by definition, FFx1_Arec and FFx1_NArec domains are essentially specific subsets of FFx2 domains. They behave like a FF domain in one record type and like a non-CDN in the other. Thus, for brevity, their plots have not been included as they would mostly be redundant.



5: IP usage for hostingprod.com (non-CDN)



6: IP usage for tsqfsny.jukutuxef.cn (MAL)

Recruitment Speed: refers to the speed (or rate) at which one observes new, unique IPs for a given domain when monitoring that domain's DNS queries over time. Based on our collected data, we have seen three major recruitment speed strategies, with each strategy being employed by a different type of domain.

Fig. 3 shows how a FFx2 domain slowly and continuously accrues unique IPs over its entire online lifetime. This behavior comes from the instability of FF domain IPs, which consist of compromised home/office computers and may go offline arbitrarily. Therefore, in an effort to help ensure reliable delivery of their nefarious content, botnets must continuously recruit new IPs. Also, compromised home computersoften obtain dynamic IP addresses from their ISP via DHCP (Dynamic Host Configuration Protocol). Consequently, a bot may be assigned different IPs over time, causing our DIGGER nodes to observe the apparent recruitment of new IPs. This effect is called *DHCP churn*, and it is not present for valid domains using stable servers with static (i.e., unchanging) IP addresses.

Meanwhile, from Fig. 4, we can see that the CDN quickly burns through most of the IPs in its more stable IP pool, achieving a much quicker recruitment speed. CDNs have a large pool of stable IP addresses, and rotate these IPs quickly and efficiently for the purpose of load balancing. They also advertise their IPs in a geographically-conscious manner. For a given CDN domain, a DNS query in Asia will often result in a diffractive set of IPs than would the same query originating in South America. This is because the CDN would mostly advertise (from its total pool of IPs) IPs located in Asia to Asian DNS servers and IPs located in South America to South American DNS servers. As a result, DIGGER's global vantage point leads us to see most of the CDN's IPs in a short period of time. In contrast, a FF domain typically can't afford the luxury of such a fine level of geographic IP management. FF domains are at the whim of the compromised computers that happen to be available and online at any given moment. Consequently, they tend to advertise the same pool of IPs irrespective of the DNS servers' geographic location. Thus, while they may change their advertised IPs as quickly as a CDN, they do so on a global scale, whereas a CDN is more localized. Therefore, our global vantage point doesn't allow us to see many more IPs than we would at any given local vantage point, causing us to observe the comparatively slower, more steady slope in the IP recruitment rate for FF domains.

Lastly, a non-CDN (shown in Fig. 5) hardly recruits any additional IPs over time. Rather, its IP pool consists of a small number of stable content servers that are almost simultaneously advertised to DNS servers around the world.

Due to the stark contrast in how these different types of domains recruit IPs, we suspect this feature will be very useful in differentiating between them.

Recruitment Period: represents the amount of time during which new IPs are seen for a given domain when monitoring that domain's DNS queries over time. A non-CDN domain, using a small pool of very stable IPs, will have almost no recruitment period; all the IPs used are advertised initially and used throughout the lifetime of the domain (as shown in Fig. 5). A CDN domain, on the other hand, uses a much larger IP pool, from which it advertises different IPs based on geographic location and load balancing. Thus, we expect CDNs to have a recruitment period. However, since CDNs have a high recruitment speed (as previously discussed) and quickly advertise most of the IPs in their IP pool, we do not expect them to have a very long recruitment period. When looking at the total online time for a CDN, we expect to see a short recruitment period at the onset of the monitoring period, followed by a longer period during which we mostly observe previously-seen IPs. This trend is clearly demonstrated in Fig. 4, and we can see that the CDN's recruitment period is smaller than the total online period of the domain. From Fig. 3, it

is apparent that the recruitment period for the FFx2 domain *old-and-girl.com* is the same as its total online period. That is, the entire time we observe the FFx2 domain to be online, it is recruiting new IPs. The constant recruitment of IPs is a result of DHCP churn and the unreliable nature of the compromised computers serving as bots. When the compromised computers go offline, they are no longer available for use in the botnet. As a result, new computers must regularly be advertised to the DNS servers to ensure the malicious content is reachable. Of course, this results in the nearly constant introduction of new IPs and the observed recruitment period. The varying recruitment periods of the different domain types should provide a beneficial metric for distinguishing between them.

E. IP Continental Distribution

Having compiled a global view of numerous domains' DNS behavior, we examined how FF domains, CDN domains, and non-CDN domains differ in respect to their IP distribution (i.e., where the IPs returned in DNS queries are located geographically). We chose to examine the geographic location of IPs based on continent instead of the more finely grained country, because we quickly discovered that, due to the close proximity of countries in Europe, a country-based resolution would be too fine. When viewing the IP distribution based on continent, however, distinguishing trends between the domain types became more apparent.

In analyzing a domain's IP distribution we asked the following questions:

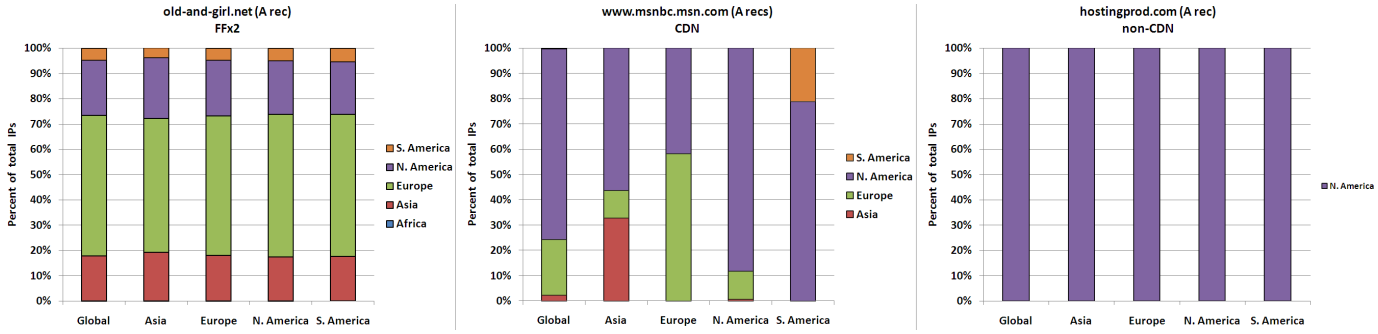
Q1: What percentage of IPs returned in the DNS queries are located in a different continent than the DNS server that was queried? We restate this, for succinctness, as the *percentage of IPs from the wrong continent*.

Q2: What percentage of IPs returned are located in each continentand based on the continent where the DNS servers being queried are located? Likewise, for succinctness, we restate this as the *continental IP distribution*.

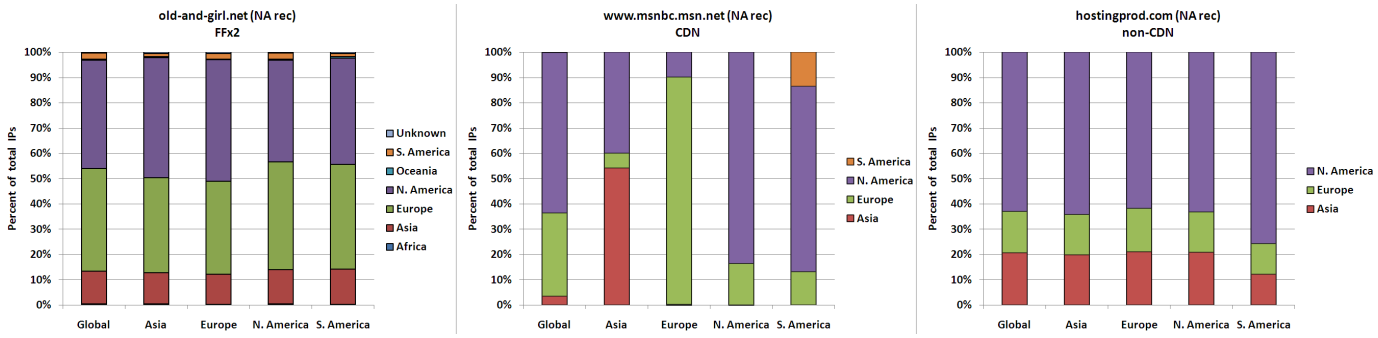
The answer to Q1 can be seen in Fig. 9 for some representative domains. For each domain, we plotted the percentage of A and NA rec IPs from the wrong continent. From Fig. 9, it is evident that the CDN domain has a considerably smaller proportion of IPs from the wrong continent than the other domain types. For both the CDN's A and NA rec IPs, the percentage from the wrong continent is less than half that of the next lowest domain. Insight into continental IP distribution (Q2) can be found in Figs. 7 and 8 for some sample domains. For brevity, we have not plotted any FFx1 domains, since their results are a subset of the FFx2 domain type. In Figs. 7 and 8, the bars represent the continental IP distribution from different perspectives. In each domain's plot, the first bar represents the continental IP distribution from a global perspective, while the other bars are from the perspective of the different continents where we deployed DIGGER nodes. For example, the bar labeled "Asia" under the *old-and-girl.com* plot in Fig. 7 indicates the percentage of A rec IPs located in each continent base on queries by DIGGER nodes in Asia to DNS servers in Asia. It is interesting to note in Figs. 7 and 8 that the continental IP distribution for both FFx2 and non-CDN domains is fairly consistent across the different continents, hardly deviating from the global distribution. For CDN domains, on the other hand, the distribution varies greatly.

The results in Figs. 9, 7, and 8 are promising. They indicate that the percentage of IPs from the wrong continent and the variance of the continental IP distribution across continents could potentially serve as features for distinguishing CDN domains from the other domain types. Furthermore, these results are in agreement with our current understanding of the various domain types. Because a goal of CDNs is to provide fast, reliable services to end users, their DNS query results often contain a majority of IPs located near the DNS server and the issued query, permitting quick content delivery by reducing the distance data has to travel. Due to this *location-aware DNS advertisement* strategy, CDNs demonstrate a smaller percentage of IPs from the wrong continent and a larger variance in continental IP distribution than other domain types. That is, more of a CDN's IPs are located in the same continent where the DIGGER nodes—and DNS servers queried—reside, with respect to other continents.

Non-CDN domains operate with a much smaller number of servers (both content and name) than CDN domains, resulting in a smaller pool of server IPs. With a smaller set of stable servers, non-CDNs don't require complicated load balancing or location-aware DNS advertisement. Instead, they adopt a form of *naive DNS advertisement* and indiscriminately advertise their small pool of server IPs around the world nearly simultaneously (as can be seen in Fig. 5). As a result, regardless of where DIGGER monitors a non-CDN domain's DNS queries, it will discover the same, relatively small, set of IPs. This causes the continental IP distribution at each continent to be the same as the global distribution. Consequently, the percentage of IPs from the wrong continent will reflect the global distribution of our DIGGER nodes, depending on the location of the non-CDN domain's servers. Fig. 7 shows that for the non-CDN domain *hostingprod.com*, almost all of the A rec IPs are in N. America. Because about 46% of



7: Percentage of total A rec IPs seen from each continent by DIGGER nodes globally and in each continent



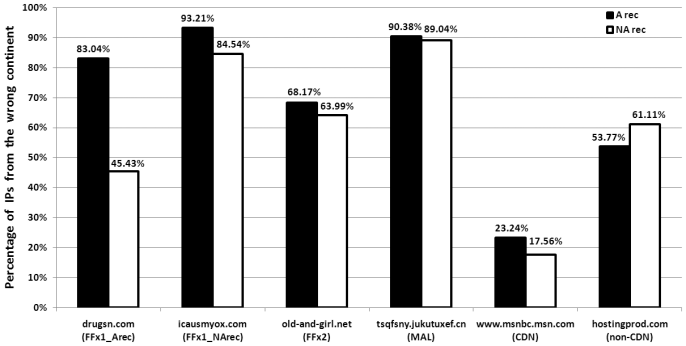
8: Percentage of total NA rec IPs seen from each continent by DIGGER nodes globally and in each continent

our DIGGER nodes are located in N. America (Fig. 2), we find that 53.77% of *hostingprod.com*'s IPs are from the wrong continent (Fig. 9), approximately the same percentage as DIGGER nodes *not* in N. America.

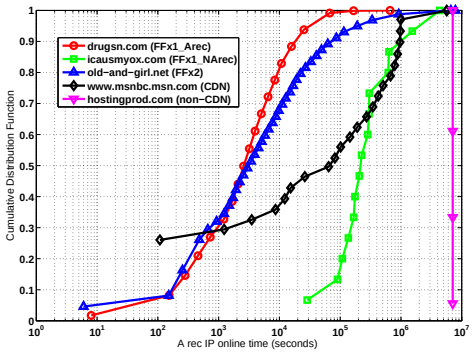
Unlike CDNs (location-aware DNS advertisement) and non-CDNs (naive DNS advertisement), FF domains use what we term *necessity-based DNS advertisement*. The advertisement strategy employed by botnets seems dictated by the unstable nature of the individual bots. Since bots can go offline at any time, FF domains must rely on whichever bots are currently available, regardless of geographic location. While FF domains don't concurrently advertise their entire IP pool globally (as non-CDNs do), they will advertise most of their IPs globally—eventually. As necessity dictates, available IPs will be advertised to DNS servers around the globe, with little or no regard to location. This is why FF domains have such a large percentage of IPs from the wrong continent and why their continental IP distribution is nearly identical, across all continents, to the global distribution. Fortunately, these features should permit us to discern FF and non-CDN domains from CDN domains. This can greatly simplify the detection of FF domains by helping identify them from CDNs, which are often very similar in other respects.

F. Other Features

1) IP Address Online Time: The IP address online time is defined as the time period during which the IP address is active (i.e., the IP address appears in the DIGGER query results). Because of the different sources of IPs used by FF, CDN, and non-CDN domains, the online time of these IP addresses should vary with their type. Both CDN and non-CDN domains host their content on well-maintained and stable servers throughout their lifetime, ensuring constant services to their customers. As a consequence, the online time of their IPs is expected to be long. FF domains, on the other hand, advertise IP addresses that primarily come from compromised computers with unreliable connectivity. Thus, the online time for FF domain IPs is usually dramatically shorter than for the IPs of valid domains. Fig. 10 shows the CDF (Cumulative Distribution Function) of A rec IP online times for both valid and FF domains. As expected, A rec IPs for non-CDN, CDN and FFx1_NArec domains have much longer online times than FFx1_Arec and FFx2 domains, which make use of compromised computers for their A rec IPs. For example, the percentage of A rec IPs with an online time less than 10^4 seconds ($\approx$2.8 hrs) was $\approx$81% for the FFx1_Arec domain, $\approx$68% for the FFx2 domain, and only $\approx$37% for the CDN domain; neither the FFx1_NArec



9: Percentage of returned IPs in different continents than DIGGER node issuing the DNS query



10: CDF for A rec IP online time

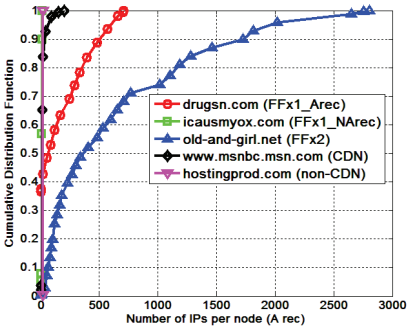
nor the non-CDN domain has any A rec IPs with an online time under 10^4 secs. Unfortunately, while the A rec IP online time appears to be a useful feature for identifying some malicious domains (i.e., FFx2 and FFx1_Arec), the same cannot be said for the NA rec IP online time. The FF domains advertise the IPs of more stable bots for their NSes, making the online time of their NA rec IPs too close to that of valid domains for classification purposes.

2) Number of Unique IP Addresses per Node: Another interesting feature is the number of unique IPs seen across the DIGGER nodes over time. To better understand this feature, we have generated CDF plots, showing the number of unique A and NA rec IPs observed by our 240 DIGGER nodes over the ≈3.5 month monitoring period.

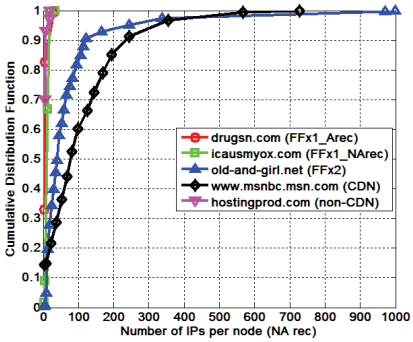
First, let's examine Fig. 11, the CDF plot for the number of unique A rec IPs per DIGGER node for some representative domains. Neither the non-CDN nor the FFx1_NArec domain had more than 18 unique A rec IPs per node. Because the domains host their content on a few, stable servers, we observe the same small set of IPs at each DIGGER node, which is independent of DIGGER node's geographic location. In the CDN domain's case, more than 90% of the DIGGER nodes observe a small number of unique A rec IPs. Specifically, ≈84% of the DIGGER nodes observed less than 22 unique A rec IPs, ≈98% observed less than 100, and no node observed more than 200. For the small percentage of DIGGER nodes that observe a slight increase, the number is still relatively small and is likely the result of load balancing. As FFx2 and FFx1_Arec domains' bots go offline over time, botmasters must continuously advertise new bot IPs with DNS to ensure the availability of their malicious content. This trend is captured by the DIGGER nodes and shown in Fig. 11. For the FFx1_Arec domain, ≈45% of the DIGGER nodes detected over 100 unique A rec IPs, more than 35% detected over 200, and a few observed over 700. The numbers observed for the FFx2 domain are even higher, with over 80% of the nodes observing more than 100, ≈63% over 200, ≈43% more than 500, and several with more than 2,500. Clearly, the FFx1_Arec and FFx2 domains possess a much higher average number of unique A rec IPs per node—a direct consequence of the bots' unreliable connectivity and, to a lesser extent, DHCP churn.

While the average number of unique A rec IPs per node appears promising as a feature for discriminating FFx1_Arec and FFx2 from other domain types, the same cannot be said for the average number of unique *NA rec* IPs, shown in Fig. 12. From the plot, it is apparent that FFx2 and CDN domains possess many more unique NA rec IPs per node than any of the other domain types, including the FFx1_NArec domain. While we expected this behavior from the FFx2 domain (for similar reasons as those described for the A rec IPs), the CDN domain's behavior came as a surprise. Although the CDN domain appears to utilize more unique NA rec IPs per node on average, the FFx2 domain does demonstrate a greater number of unique IPs seen at a single node: 999 IPs to the CDN domain's 727. It seems that, over time, CDNs can advertise numerous NSes with DNS, resulting in an excessive number of unique NA rec IPs per node. This behavior could arise because the CDN is trying to ensure the availability of its NSes, affording it control to perform load balancing. In any case, the behavior of the FFx2 and CDN domains is very similar, causing the number of unique NA rec IPs to be an indistinctive feature.

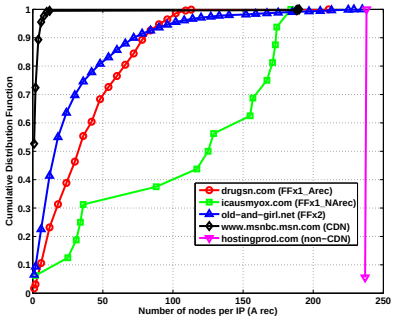
3) Number of Nodes per IP Address: Since the number of unique IPs per node proved a promising feature for differentiation, we decided to look at the relationship from the inverse perspective: for individual IP addresses, how many different nodes (i.e., DNS servers) were the IP addresses observed on. We restate this as the *number of nodes per IP address*. For the NA recs, this feature demonstrated no useful trends for differentiation, but some interesting behavior emerged in the A recs. Plotting the CDF for the number of nodes per A rec IP (Fig. 13), we can clearly



11: CDF: #of unique A rec IPs per DIGGER node



12: CDF: # of unique NA rec IPs per DIGGER node



13: CDF: # of nodes per A rec IPs

see that FFx2 and FFx1_Arec domains exhibit remarkably similar trends, separate from those of the other domain types. Since the non-CDN domain advertises its small set of stable A rec IPs with every DNS server around the globe, we observe nearly all the IPs on every node, and thus a large number of nodes per IP. The FFx1_NArec domain behaves similarly. However, since some of its A rec IPs might belong to bots (or otherwise unstable content servers), some of its IPs may not appear on all nodes. As a consequence of its location-aware DNS-advertisement strategy, many of the CDN domain's IPs will only be advertised to a small set of nearby DNS servers, keeping its average number of nodes per IP small. Likewise, the FFx2 and FFx1_Arec domains fall somewhere in between due to their necessity-based DNS-advertisement strategy. It may be the case that bots with unstable connectivity only get advertised to a handful of DNS servers before they go offline. If a bot only intermittently loses connectivity, its IP may eventually propagate to more DNS servers, increasing its nodes-per-IP count. However, if the bot permanently disconnects from the botnet, its nodes-per-IP count will remain stunted, decreasing the overall average nodes-per-IP.

4) Total Number of Unique IPs: The total number of unique IPs seen across all nodes over time proves remarkably apt as a metric for distinguishing non-CDN domains from CDN and FF domains. This is because, unlike CDN and FF domains, non-CDN domains advertise only a few stable content and name servers with DNS. Since a non-CDN domain's A and NA rec IPs are seen ubiquitously around the globe, the total number of unique IPs observed by the DIGGER nodes over time will be meager. Table I, which shows the number of IPs in the A and NA recs for examples of the different domain types, demonstrates this effect. The CDN and FFx2 domains display abundant IPs in their A and NA recs. While the FFx1_Arec domains possess a modest number of NA rec IPs, they have a substantial number of A rec IPs—a clear distinction from a non-CDN domain. The opposite holds true for the FFx1_NArec domain; the small number of IPs cause its A rec to resemble a non-CDN domain, while the much larger number of NA rec IPs betrays this guise.

5) Reverse DNS Lookup and TTL: The last two features we will discuss seem to be obvious candidates for use in classification: the reverse DNS lookup result and the TTL values of the A and NA recs. Clearly, if the reverse DNS lookup on a domain contains suspicious words typical to home computers (e.g., comcast, dynamic, dial-up, etc.), it is a strong indicator that the IP belongs to compromised computer, or bot. Because an IP's reverse DNS name is set by the IP's service provider and not the owner of the domain, it cannot be faked by a botmaster. This makes it a fairly useful metric for identifying bots. Unfortunately, the reverse DNS lookup is highly unreliable. Often, a reverse DNS lookup will not return a result, thus providing no insight into the actual identity of the suspect IP. Additionally, we don't have a complete list of suspicious words, and occasionally, the presence of such words may not be indicative of a bot; often, it is only after thoroughly analyzing the DNS data in conjunction with the reverse DNS words that we can determine them to be bad, strengthening a malicious classification. Therefore, we have decided not to incorporate the reverse DNS name for automatic domain classification. Instead, when present, we use it to help reinforce or confirm our manual identification of the different domain types. By omitting it from our automatic identification, we hope to gain a better insight into potential of the more reliable classification features.

The A and NA recs' TTL values also appear highly useful for differentiating between the domain types. CDNs and FF domains tend to use small TTL values, affording them a high level of control over the domain's IPs. CDN domains use this extra control for load balancing and reliable content delivery. FF domains are really only concerned with reliable content delivery in the presence of unreliable content servers (i.e., bots). Non-CDNs, unperturbed by

these concerns, use much longer TTL values for their stable content and name servers. However, unlike many of the other features we have previously explored, the TTL value is not an uncontrollable consequence of a botnet. While it is difficult for a botmaster to mimic features such as a CDN's location-aware DNS-advertisement strategy or a valid domain's recruitment speed/period without sacrificing content availability, this is not the case with the TTL value. An IP's TTL value is set by the owner of the domain. A botmaster can easily increase the average TTL value for its A or NA records without sacrificing the availability of the malicious content. By setting a short TTL value for some IPs and very large TTL values for others, the average TTL of a FF domain can be made to look like the average TTL of a non-CDN domain, without sacrificing the fine level of control over some of the IPs. Those IPs with large TTLs (used to inflate the average value) could belong to more reliable bots; they could just as easily be bogus IPs that don't resolve to anything. So long as some of the IPs (presumably those with the shorter TTLs) resolve to online bots, the malicious content can still be reached. While we could try more complicated methods of measuring the TTL values to account for this inflation technique, it would be just as easy for botmasters to come up with another clever way to circumvent our metric. Botmasters simply have too much control over the TTL value for it to be a reliable feature for classification. Therefore, we have decided not to use it as such. It should be noted that other features, like the recruitment speed and period, cannot be as easily manipulated by the botmaster, since the unstable bot IPs necessitate constant recruitment.

G. MAL domains

As previously discussed in Section III, a MAL domain falls somewhere on the spectrum between a non-CDN and FF domain. It is certainly less stable (over time) than a non-CDN domain, but it is not fluxy enough in its A or NA records to be considered a FF domain. While it may utilize stable bots for its content or name servers, it most likely employs a stable server rented—or possibly hijacked—from a hosting provider. In this sense, it is similar to a non-CDN domain. Yet, unlike a non-CDN domain, a MAL domain is not benign. It is a malicious domain, partaking in malicious activities. As a consequence, its IPs will likely be blocked eventually, requiring it to register fresh IPs with DNS in order to maintain its content availability. Therefore, assuming it will be eventually detected and blocked, it must slowly and continuously recruit new IPs—albeit much more slowly than any FF domain.

This DNS advertising behavior means that, like FF and non-CDN domains, MAL domains will exhibit a large percentage of IPs from the wrong continent. This trend is shown for a representative MAL domain (*tsqfsny.jukutuxef.cn*) in Fig. 9. Likewise, MAL domains will demonstrate a much smaller variance in their continental IP distribution across continents than CDN domains, although we have neglected this plot due to space constraints. As a result, these two features should still allow CDNs domains to easily be identified from the other domain types.

Other interesting features worth discussing for MAL domains include the total number of unique IPs, the IP overlap, and the recruitment speed and period. As can be seen from Table I, while the representative MAL domains (*duelreal.com* and *tsqfsny.jukutuxef.cn*) have a small number of total unique IPs (like a non-CDN domain), their IP overlap is exceptionally high (like a FF domain). Almost all of their A rec IPs are also used for their NA recs. This sets them apart from both non-CDNs and FF domains, providing a useful metric for classification. Looking at Fig. 6, we can see that the MAL domain *tsqfsny.jukutuxef.cn* demonstrates a slow and steady recruitment of IPs. Clearly, this is different than the recruitment behavior of a non-CDN domain (Fig. 5); however, it initially appears quite similar to that of a FF domain (Fig. 3). Upon closer examination, it is revealed that unlike FF domains, which recruit hundreds to thousands of IPs, the MAL domain recruits only tens of IPs over $\approx$3.5 months.This is a drastic difference, and it should prove beneficial in distinguishing MAL domains from non-CDN and FF domains.

V. Detection Methodology

A. Overview

Our observations in Section IV indicate that the different domain types could be identified based on behavioral features of their global DNS activity. To demonstrate this, we have build a a rudimentary, proof-of-concept detector, utilizing a multi-leveled linear SVM (Support Vector Machine) classifier. The rest of this section describes the design and implementation of this classifier, including how we quantified the behavioral features, chose which features to apply at each stage (or level), determined the order of the stages, and finally, how the SVMs were trained.

Classification Feature	DNS Record Type	Domain Type Classification Groups
F1. Avg. # of unique IPs per Node	A	• [CDN_1, non-CDN_2, MAL_3, FFx1_NArec] • [$FFx2_4$, FFx1_$Arec_5$]
	NA	• [CDN_1, $FFx2_4$] • [FFx1_NArec] • [non-CDN_2, MAL_3, FFx1_$Arec_5$]
F2. Avg. # of Nodes per IP	A	• [CDN_1] • [non-CDN_2] • [FFx1_NArec] • [MAL_3, $FFx2_4$, FFx1_$Arec_5$]
F3. A & NA rec overlap	A & NA	• [CDN_1, non-CDN_2] • [MAL_3, $FFx2_4$, FFx1_$Arec_5$, FFx1_NArec]
F4. % IPs from wrong continent	A	• [CDN_1] • [non-CDN_2, MAL_3, $FFx2_4$, FFx1_$Arec_5$, FFx1_NArec]
	NA	
F5. Continental IP distribution's average cosine similarity	A	
	NA	

The number by each domain type represents the level it is classified by our SVM. When selecting features for SVM- ***x*** *, domains with a number <* ***x*** *can be ignored, since they have already been classified and removed from the unknown set.*

Classification Feature	DNS Record Type	Domain Type Classification Groups
F6. IP recruiting speed	A	• [CDN_1] • [non-CDN_2, FFx1_NArec] • [MAL_3] • [$FFx2_4$, FFx1_$Arec_5$]
	NA	• [CDN_1] • [non-CDN_2, FFx1_$Arec_5$] • [MAL_3] • [$FFx2_4$, FFx1_NArec]
F7. IP recruiting period	A	• [CDN_1] • [non-CDN_2, FFx1_NArec] • [MAL_3] • [$FFx2_4$, FFx1_$Arec_5$]
	NA	• [CDN_1] • [non-CDN_2, FFx1_$Arec_5$] • [MAL_3] • [$FFx2_4$, FFx1_NArec]
F8. Total unique IPs	A	• [CDN_1, $FFx2_4$, FFx1_$Arec_5$] • [non-CDN_2, MAL_3, FFx1_NArec]
	NA	• [CDN_1, $FFx2_4$, FFx1_NArec] • [non-CDN_2, MAL_3, FFx1_$Arec_5$]
F9. Avg. IP online time	A	• [CDN_1, non-CDN_2, FFx1_NArec] • [MAL_3] • [$FFx2_4$, FFx1_$Arec_5$]

II: Features for classifying the domain types into different groups

B. Classification Features

Table II shows the features we considered using in the classifier and how they are likely to group the domain types. Each feature has been given a number to simplify its representation throughout the paper. With the exception of feature F3, each feature can be applied to a domain's A or NA rec, and while not displayed in Table II, the features can also be applied to the combined IP pool of the A and NA recs, represented as (A + NA). Notice that we do not consider the NA rec for features F2 and F9, because our analysis showed that they were not useful distinguishing features. Lastly, the column labeled "Domain Type Classification Groups" in Table II shows how each feature—when applied to the A or NA rec—will likely group the different domain types, represented by square brackets. Table II does not express hard-and-fast rules for how features classify the domain types. Rather, it shows *likely* groupings: domain types tending to produce similar results with respect to a given feature and record type. Thus, Table II is a helpful visual tool for determining the application of features at different SVM levels. Using numerical subscripts, we have indicated the order our classifier detects the domain types.

1) Spatial and Temporal Incongruities: As previously mentioned, DIGGER collected DNS data from around the world on over 106,000 different domains for ≈3.5 months. During that time, some of the PlanetLab nodes sporadically went offline. This could result from a number of possibilities, including node maintenance, improper configuration, node failure due to over-utilization, etc. As a result of this instability, our data contains some gaps in its spatial consistency: sometimes, we are missing data from different parts of the world. Compounding this problem, is the temporal inconsistency introduced by the nature of malicious domains. When it is discovered that a domain is partaking in malicious activities, DNS servers may choose from a couple of countermeasures. Some may choose to block (or blacklist) the domain, responding that the domain is unknown or doesn't exist. Another option is to perform DNS domain IP parking, replying with an IP address that doesn't belong to the malicious domain—possibly belonging to a website informing the user that the domain is unreachable or has been blocked. Not only do DNS servers handle identified malicious domains differently, they may do so at different times, or not at all. When taken together with the spatial inconsistency introduced by instability of the PlanetLab nodes, we find that DIGGER doesn't have a complete global view for certain domains. In most cases, this means we are only missing data from a few nodes around the globe at any given time. Considering the large number of nodes we gather data from, the effect is negligible. However, in the worst cases, we only have a handful of nodes that managed to gather relevant

DNS data for a domain before it's taken offline and replaced by its owner. In these worst-case scenarios, our view might be confined to just a few countries or continents. While many of the features in Table II are robust in the presence of spacial inconsistencies, F2 is not. For this reason, we have chosen not to use it in our data set, although it could still serve as a reliable metric for classification. We have also decided to omit the temporally-sensitive feature, F9. To effectively use such a feature, one should first determine the optimal monitoring period for detection and then rigorously monitor each domain for that specific period of time. Otherwise, the temporal deviations caused by malicious domains going offline become influential. One could monitor a malicious domain only at the tail-end of its lifetime while monitoring another from its onset to its demise. Both are malicious domains, yet they would have very different average IP online times simply as a consequence of *when* in their lifetimes they were monitored. How to solve this problem and that of finding an optimal monitoring period despite domains unexpectedly going offline, DNS domain IP parking, and failing nodes is beyond the scope of this paper. Furthermore, it is unnecessary since we can rely on other features which are more resilient to temporal deviations. By neglecting F9, we can build our classifier to operate over our entire data set, spanning ≈3.5 months.

Neglecting features F2 and F9 is reasonable for a proof-of-concept classifier. Since our main goal is to demonstrate the potential usefulness *most* of these differentiating features possess for classification, we leave the problem of finding the absolute minimal monitoring period and number of monitoring nodes (and their location) as future work.

2) Feature Quantification: With the exception of F2 and F9, which we don't use for reasons previously explained, the features in Table II are quantified as outlined below. All of the features, except F3 (A & NA rec overlap), were quantified using the IPs of the three different record types—A, NA and (A + NA) recs—to produce 3 distinct values. Which of these values is used at each stage of the classifier is discussed in Section V-C.2. Each feature is calculated for each domain monitored by DIGGER over the total ≈3.5 month duration.

F1: Let P_i = number of unique IPs on node i, and let N = number of nodes (of the 240 total) where the number of unique IPs ≥ 1. Then, the average number of unique IPs per node (F1) is computed as:

$$\text{F1} = \frac{\sum_{i=1}^{N} P_i}{N} \tag{1}$$

F3: represents the percentage of unique IPs that overlap between the A and NA recs. Thus, if all the IPs from one record type are also used for the other record type, there will be a 100% IP overlap. For a given domain across all nodes, let P_A be the set of unique A rec IPs and P_{NA} be the set of unique NA rec IPs. Then, F3 is calculated as:

$$\text{F3} = \frac{|P_A \bigcap P_{NA}|}{\min\{|P_A|, |P_{NA}|\}} \tag{2}$$

F4: Using an online database [25], we were able to determine the country of origin for most IPs observed by DIGGER. For those IPs not present in the database, we were able to perform a "who is" lookup and determine most of their countries of origin. The few remaining IPs whose location couldn't be determined were labeled "unknown". Thus, for nearly all IPs monitored by DIGGER, we could determine which continent the IP was located on: N. America, S. America, Europe, Asia, Africa, Oceania, Antarctica, and—very rarely—unknown. Let W_i = number of unique IPs on node i that are located in a different (i.e., wrong) continent than node i. Let P_i = total number of unique IPs on node i. Then, the percentage of IPs from the wrong continent (F4) is computed as:

$$\text{F4} = \frac{\sum_{i=1}^{N} W_i}{\sum_{i=1}^{N} P_i}. \tag{3}$$

F5: We want to determine the *average* continental IP distribution across all nodes from a given continent. To obtain this, we grouped the nodes together based on the continent they are located in. Then, we examined each group of nodes, tallying the number of unique IPs (per node) seen from each continent. If, for example, an IP appears on more than one node from a given continent, it will be counted once for each node it appears on. Calculating a continent's continental IP distribution in this way is more robust to misbehaving or abnormal nodes and better reflects the continental IP distribution of the majority of nodes from a given continent.

Recall from Section IV-E that CDN domains differ from the other domain types due to their location-aware DNS advertisement strategy. The continental IP distribution of a CDN domain will be biased in favor of the queried node's continent. Contrarily, the other domain types will demonstrate nearly identical continental IP distributions

regardless of the queried node's location (see Figs. 7 and 8). Therefore, we want to quantify how similar this distribution appeared between continents, enabling us to discern CDNs from the other domain types.

Let the continents N. America, S. America, Europe, Asia, Africa, Oceania, Antarctic and "unknown" be represented by the numbers 1–8, respectively. Then, n_i = number of nodes on continent i, for $1 \leq i \leq 4$ (continents with DIGGER nodes). For node j, let $\hat{a}^j$ be a vector representing the number of unique IPs seen from each continent. Thus, $\hat{a}_i^j$ is the number of unique IPs from continent i that were seen on node j. Then, for each continent i with DIGGER nodes, where $1 \leq i \leq 4$, we calculate $\hat{A}^i$ as shown in Eq. (4). We calculate the cosine similarity (shown in Eq. (5)) between every possible pair of vectors $\hat{A}^i$, for $1 \leq i \leq 4$, and then take the average, producing the IP continental distribution's average cosine similarity (F5). The closer this value is to 1, the more similar the continental IP distributions appear on each continent, and the less likely the domain is a CDN domain.

$$\hat{A}^i = \sum_{j=1}^{n_i} \hat{a}^j \qquad (4) \qquad \text{Similarity}(\hat{X},\hat{Y}) = cos\theta = \frac{\hat{X} \bullet \hat{Y}}{\| \hat{X} \| \| \hat{Y} \|} \qquad (5)$$

F6/F7: First, we calculate a domain's online time, denoted as T_o, as the amount of time we consider the domain to be online. Analyzing all available DNS query data from all nodes, we consider an *online point* to be a point in time where we have observed IP addresses. If the difference in time between two consecutive *online points* is less than a threshold of several hours, we add it to the T_o. Next, we calculate the domain's recruit time, denoted as T_r. We consider a *recruit point* to be a point in time where we have observed a *new* IP address (i.e., one that hasn't occurred earlier in time). If the difference in time between two consecutive *recruit points* is less than the threshold, we add the it to T_r. Let $P =$ the total number of unique IPs observed globally for a domain. Then, the IP recruiting speed (F6) and period (F7) are calculated as:

$$\text{F6} = \frac{N}{T_r} \qquad (6) \qquad \text{F7} = \frac{T_r}{T_o}. \qquad (7)$$

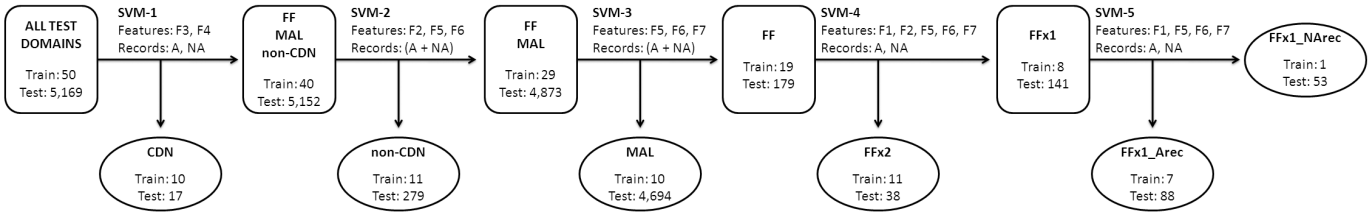
In those instances where all of a domain's IPs are observed instantaneously, resulting in a $T_r = 0$, we set F6 to 1. This value corresponds to a rate of one new IP every second, and it was great enough in magnitude from all other observed values to serve as a rough approximation for infinity.

F8: We look at every DNS query gathered by all the DIGGER nodes. Whenever we encounter a previously-unseen IP, we count it. After examining all available DNS records, the final sum is considered the total unique IPs (F8) for a domain. It represents the number of different IPs used by a domain around the world.

C. SVM Classifier

1) Rule-based Filter: Before testing our SVM classifiers, we applied a simple, rule-based filter to remove any domains that were unlikely to be malicious. The filter also ignores domains that clearly belong to CDNs, allowing us to test the accuracy of our SVM detector. If any of the following rules applied to the domains, they remained in the testing set, otherwise they were removed: (1) any IP in its A or NA rec had a max TTL less than 1 day, (2) its A or NA recs contained more than 10 IPs over the entire monitoring period, (3) its reverse DNS lookup contained a suspicious word (e.g., comcast, charter, dynamic, dialup, etc.), and (4) its reverse DNS lookup indicated it was a known CDN domain (e.g., contained words like akamai). This simple filter removed all the valid, easily identified domains. Any domain with a max TTL value of more than a day in both its A and NA recs is probably not suspicious. If it is FF domain or a MAL domain using stable servers and acting sufficiently suspicious (i.e., its IPs are becoming blocked), it should accrue more than 10 IPs after ≈3.5 months of monitoring. Clearly, if any of its DNS lookups indicate the use of a home computer it could be malicious, warranting further examination. Lastly, any domains with reverse DNS lookups indicating known CDNs are included so we can test our SVM's ability in identifying CDN domains. Applying this filter to our set of 106,000+ domains reduced our testing set to 5,422 remaining domains. Finally, we removed any domains with insufficient DNS query data. This included 250 domains momentarily observed by single nodes and 3 domains monitored by less than 25% of our DIGGER nodes, bringing the total testing set to 5,169 domains.

2) Multi-level SVM: Fig. 14 shows the design of our multi-leveled SVM classifier and the results of our training and testing sets. Each level of the SVM classifies one of the domain types from the total set of unknown domains. This progressively reduces the number of unknown domains at each level, simplifying the task at subsequent levels and allowing us to automatically identify the domain types. Each oval in the figure represents a domain type that has been classified. Each rectangle represents a set of multiple, unknown domain types remaining to be classified.



14: SVM flowchart

	b	F1		F2	F3		F4		F5			F6			F7			Result > 0
	(bias term)	A	NA		A	NA	A	NA	A	NA	(A + NA)	A	NA	(A + NA)	A	NA	(A+NA)	
SVM-1	284.69				-108.10	-88.90	-124.83	-160.60										*CDN*
SVM-2	128.26			-217.20							47.23			-2,072.45				*non-CDN*
SVM-3	192.04										1.32E-06			-4.06E-03			-1.77	*MAL*
SVM-4	-390.75	3.13	12.54	0.27					-1.14E-03	-0.03		0.42	0.21		-0.37	0.38		*FFx2*
SVM-5	933.52	0.42	-0.03						2.28E-06	0.02		7.30E-04	-4.01E-03		1.74	-5.31		*FFx1_Arec*

III: Linear SVM equations

The values for "Train" show how many examples of a given domain type (or group of domain types) were used when training that level of the classifier. The values for "Test" indicate the number of domains that were classified (or remained to be classified) when we applied each tier of the classifier to our testing set. We manually identified about 10 representative domains of each type to be used in training, as show in in Fig. 14. More difficult to detect by hand, we were only able to manually identify a single FFx1_NArec domain.

Table III shows the bias and feature weights for each level of our classifier. Those features not used at a particular level are shaded black. For each SVM, the *Result* is calculated as the *bias term* plus the product of the feature and its weight. The "*Result* > 0" column indicates how a domain with a positive *Result* will be classified. The exception is FFx1_NArec domains, which are classified when SVM-5's *Result* is negative. In addition to indicating how the domain should be classified, the magnitude of the *Result* represents the confidence in classification choice.

As we classify each domain type, it is removed from the set of unknown domains before applying the next SVM level. Thus, when considering the classification features for level *SVM-x*, we can ignore domain types in Table II with numbers less than *x*. Due to the similarities some domain types share between certain features, the *order* we apply the classifiers and which features we use at each level becomes important. The proper order can exploit the strong differentiating features between certain domain types. We will now explain the features used at each level of our SVM classifier and justify the order of classification.

SVM-1: CDN domains tend to have a short recruit period (F7) and a fast recruit speed (F6) when compared to MAL and FF domains. In the case of non-CDN domains, all the IPs are often seen simultaneously, resulting in no recruit period and an instantaneous recruit speed. Since MAL and non-CDN domains are similar in the total number of unique IPs seen, this difference in recruit speed and period becomes an important differentiating feature. If we were to classify non-CDN domains first, F6 and F7 would receive less weight, putting the burden of differentiation on F3 (IP overlap). Moreover, F4 and F5 are strong indicators of CDN domains due to their DNS strategy; none of the other domain types display this location-aware behavior. Therefore, we can remove CDN domains from the unknown set first with high accuracy. Since CDN domains can behave similarly to FF domains in other respects (e.g., large number of IPs), removing them first will improve successive classification. For these reasons, SVM-1 was trained on 10 CDN domains and 40 other domains (i.e., non-CDN, MAL, and FF), using F4 and F5 on the domains' A and NA recs. As we can see from Table III, a large percentage of IPs from the wrong continent (F4) or similar IP distributions on each continent (F5) will generate a negative *Result*. Thus, only CDN domains, practicing a location-aware DNS advertisement strategy, will obtain positive values. We ran SVM-1 on our testing set of 5,169 domains. It identified a total of 17 CDN domains, which we manually verified then removed from the testing set.

SVM-2: With CDN domains removed from the testing set, F6 and F7 could now be used to their full potential. While non-CDN domains advertise all there IPs nearly instantaneously, both MAL and FF domains will need to recruit IPs over time. Additionally, MAL and FF domains may possess IP overlap; this should never be the case

for valid non-CDN domains. Thus, for SVM-2, we use F6, F7, and F3. However, unlike SVM-1, where we applied the features to the A and NA recs individually, SVM-2 looks at the combined (A + NA) recs, accounting for FFx1 domains demonstrating fluxy behavior in only a single record type—the other often appearing benign. We trained SVM-2 on 11 representative non-CDN domains and 29 of the FF and MAL domains. When applied to the remaining 5,152 unknown domains, it classified 279 as non-CDN. We manually analyzed the 69 boarder cases with *Results* closest to 0 and found them to be satisfactorily classified; these results will be discussed further in Section V-D.1. From Table III, we can see that F7 is the dominating feature. If the domain demonstrates any significant recruitment period, it is unlikely to be a non-CDN domain. Had CDN domains not been previously classified and removed, this feature would have been less prominent, forcing the classifier to depend on the more unreliable F3.

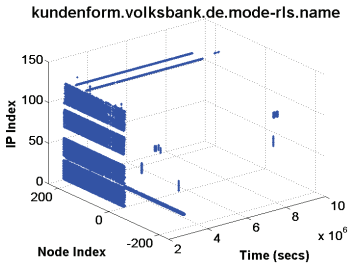
SVM-3: After removing the non-CDN domains identified by SVM-2, the testing set was entirely composed of malicious domains (i.e., FF and MAL). Due to the many similarities between FFx1 and FFx2 domains, it seemed logical to classify MAL domains next. F8 is the most obvious distinguishing feature between MAL and FF domains, but we suspected that F6 and F7 might also prove useful, since FF domains should recruit more IPs over a greater percentage of their online time. SVM-3 applies F6, F7 and F8 to the domains' (A + NA) recs, again to account for FFx1 domains. We trained SVM-3 on a representative set of 10 MAL domains and 19 FF domains. When applied to the testing set of 4,873 malicious domains, it identified 4,694 MAL domains and 179 FF domains. Looking at SVM-3 in Table III, we see that the dominant feature in distinguishing MAL domains from FF domains is F8: the number of unique IPs. Because of their slower IP recruitment rate, MAL domains will be quickly outpaced by FF domains, resulting in a much lower number of unique IPs. This difference will be accentuated with time, causing it to be the dominant classification feature for our $\approx$3.5 months of data.

SVM-4: After three stages of the classifier, only FF domains remained in the testing set. By definition, the only thing distinguishing the FF domains is which record type demonstrates fluxiness. A combination of the two FFx1 domain types, FFx2 domains should be the next candidate for classification. From Table II, it appears that applying F1, F3, F6, F7 and F8 to the individual A and NA recs should discern FFx2 from FFx1 domains. For F1, F6, F7 and F8, all the FF domains will demonstrate fluxy behavior, but the FFx2 domain will demonstrate twice as much as either FFx1 domain. This will also cause the IP overlap (F3) experienced by FFx2 domains—which use botnets for both record types—to be considerably larger. We trained SVM-4 on a representative set of 11 FFx2 domains and 8 FFx1 domains. While F6 appears less significant, features F3, F7, and F8 contribute nearly equally in classification, and F1 is a strong indicator of FFx2 domains. These results and their implications will be detailed in Section V-D.3. Applying SVM-4 to the 179 remaining FF domains resulted in the classification of 38 FFx2 and 141 FFx1 domains, which we manually verified.

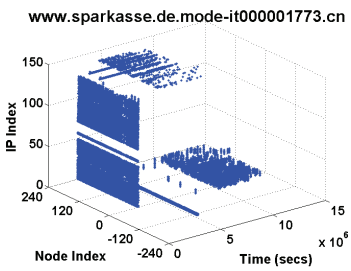
SVM-5: The final level of the classifier is charged with the modest task of discriminating between FFx1_Arec and FFx1_NArec domains. With the exception of F3, SVM-5 makes use of the same features and record types as SVM-4 for similar reasons. F3 is ignored at this stage since the FFx1 domains should experience comparable, modest-to-no IP overlap. If a FFx1 domain demonstrates too much IP overlap, the fluxy behavior becomes visible in both record types, and the domain can be considered FFx2. The usefulness of the other features is straightforward: for FFx1_Arec domains, the features will appear more fluxy in the A recs, and the opposite holds for FFx1_NArec domains. Unfortunately, we were only able to find a single FFx1_NArec domain by hand for training purposes. When applying SVM-5 to the 141 FFx1 domains, we were surprised to find 53 of them were actually classified as FFx1_NArec domains. We examined the results by hand and discovered they were indeed correctly identified as FFx1_NArec domains. We will examine these results and possible explanations in in Section V-D.4. Table III shows that F6 and F7 became negligible for SVM-5. F1 holds some influence in classification, but the dominating feature is clearly F8. By this SVM stage, the testing set consisted entirely of FFx1 domains, and since the fluxy record type naturally accrues more IPs with time, F8 strongly influences classification.

D. Results

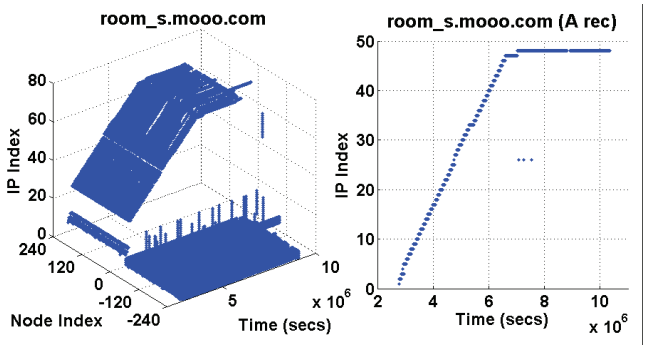
1) False Positives: From our classifier's results at each stage, only SVM-2 was found to experience any false positives; two FFx1_Arec domains were incorrectly identified as non-CDN due to DNS domain IP parking, which caused the IPs to resemble the stable and benign behavior characteristic of non-CDN domains. When we initially analyzed DIGGER's data, we discovered a couple of nodes that reliably partook in IP parking using the same set of IPs. Their parking behavior is easily observed in Figs. 15 and 16 as two long, constant lines with positive Node



15: FFx1_Arec domain: correctly classified



16: FFx1_Arec domain: misclassified as non-CDN

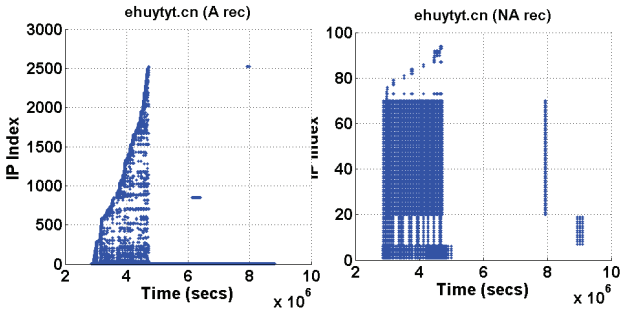


17: Cautious MAL domain

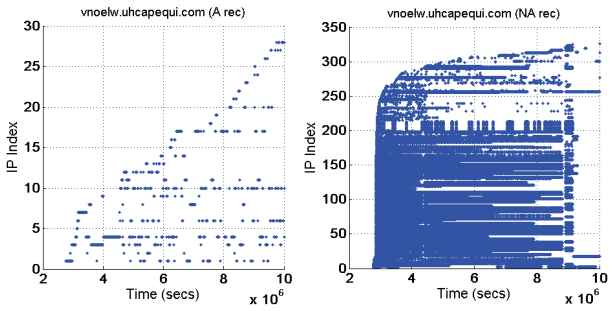
Index values, indicating parking in the A rec. Appearing as consistent, stable IP addresses, these parked IPs cause a domain to appear more benign than it actually is, and if their influence dominates, our classifier could consider the domain to be non-CDN. We removed the influence of IP parking due to these two nodes by ignoring the associated parking data when present. However, in reality, these were not the only nodes performing IP parking—though they were the most consistent. Since we didn't filter this behavior for all nodes, they affected classification, accounting for SVM-2's two false positives. For example, consider the similar domains in Figs. 16 and 15. For the misclassified domain in Fig. 16, a large majority of nodes instigated IP parking in both record types, confusing our classifier. While initially the domain appears fluxy, the parking behavior of multiple nodes dominates over its lifetime, causing it to be classified as non-CDN. While considered a false positive, this labeling is rather subjective, since for the majority of the domain's lifetime it *does* resemble a non-CDN due to IP parking. Since our classifier is temporally naive (we consider all available data over our ≈3.5 month monitoring period), this misclassification is entirely reasonable; nevertheless, it would be better to determine an optimal monitoring period and identify IP parking techniques. This is part of our future work.

2) Cautious MAL domains: While manually validating SVM-3's results, we discovered 4 borderline MAL domains exhibiting atypical IP behavior, one of which is shown in Fig. 17. Recruiting less than 50 A rec IPs over ≈2.5 months (the domain was parked afterwards), it is not fluxy enough to be considered a FFx1_Arec domain. However, its uncannily regular IP recruitment distinguishes it from other MAL domains. Further analysis revealed that the domains advertise only a single A rec IP per query, with a max TTL of one minute. Despite this fine level of control, the domains only replace the IP once a day, adhering to a meticulously precise schedule. Additionally, we can see from Fig. 17, that once changed, the A rec IPs are not reused. Since these malicious domains are not fluxy enough to be considered FF, they are correctly classified as MAL domains, but their behavior implies a management strategy different from most MAL domains. They appear to be a type of *cautious* MAL domain, regularly and preemptively replacing their A rec IPs before they can be detected and blocked—though the short TTL permits rapid response when required. With only 4 instances observed, this behavior is currently very rare. Nevertheless, the strategy is interesting and may gain popularity among malicious domain owners trying to evade current detection technologies, warranting future research into these domains and how to better detect and subvert them.

3) FF domains: Another interesting aspect of our classifier is how it distinguishes between the various FF domains. Recall from Table III that F1 is the dominant feature for SVM-4, with the NA rec being 4x as influential as the A rec. This assessment makes sense and is in agreement with our observed data. From Table I and Fig. 3, we see that the FF domains recruit more IPs for their A recs than their NA recs, making the A recs appear more fluxy. Therefore, for SVM-4, behavior that isn't considered fluxy enough for the A rec could be sufficient when demonstrated in the NA rec. The consequence of this asymmetric weighting of fluxiness can be witnessed in Fig. 18 (a domain classified as FFx2) and Fig. 19 (a domain classified as FFx1_NArec). The first thing to notice about both of these domains is that they demonstrate definite fluxy behavior in one of their record types. Fig. 18 is clearly fluxy in the A rec, while Fig. 19 is clearly fluxy in the NA rec. However, at a first glance, neither domain appears overly fluxy in their other record. The FFx2 domain seems relatively stable for most of its NA rec, with what appears to be fluxy behavior for ≈20–30 of its NA rec IPs. In the case of the FFx1_NArec domain, which only has about 30 IPs in its A rec, the recruitment behavior resembles that of a MAL domain; it slowly and consistently recruits a



18: Classified FFx2 domain



19: Classified FFx1_NArec domain

Domain Type	# of domains	% of ALL	% of ALL (TEST)	% of MAL/FF	% of FF	% of FFx1
ALL	106,311					
SIMPLE FILTER	101,142	95.14%				
ALL TEST	**5,169**	**4.86%**				
CDN	17	0.02%	0.33%			
non-CDN	279	0.26%	5.40%			
MAL/FF	**4,873**	**4.58%**	**94.27%**			
MAL	4,694	4.42%	90.81%	96.33%		
FF	**179**	**0.17%**	**3.46%**	**3.67%**		
FFx2	38	0.04%	0.74%	0.78%	21.23%	
FFx1	**141**	**0.13%**	**2.73%**	**2.89%**	**78.77%**	
FFx1_Arec	88	0.08%	1.70%	1.81%	49.16%	62.41%
FFx1_NArec	53	0.05%	1.03%	1.09%	29.61%	37.59%

IV: Relative distributions of the various domain types

small number of IPs over the duration of the monitoring period. In addition, the IP overlap for the FFx1_NArec domain is less than 4%. Thus, in this case, the classifier seems to have performed correctly: a domain with FF behavior in its NA rec and MAL behavior in its A rec should be considered a FFx1_NArec domain. However, it isn't immediately obvious why the FFx2 domain is considered fluxy in its NA rec. We already know that NA recs require less fluxy behavior to be considered FF. Clearly, the FFx2 domain does demonstrate some FF behavior in its NA rec. Furthermore, the FFx2 domain has an IP overlap of ≈26%, about the same number of NA rec IPs demonstrating recruitment behavior. Thus, ≈26% of NA rec IPs are also present in the A rec, and their fluxy behavior influences the NA rec's behavior. Because the total number of unique NA rec IPs is approaching 100 and ≈26% of them demonstrate fluxy behavior, the less stringent fluxiness demands for the NA rec are met. Since both the A and NA rec behave reasonably fluxy, the domain is correctly classified as FFx2.

4) Domain Type Distribution: Table IV shows the number and distribution for each domain type identified by our classifier. For example, it shows that of the 106,311 domains we monitored, our rule-based filter (Section V-C.1) identified 101,142 domains as benign or lacking in sufficient data—corresponding to 95.14% of our monitored domains. This is reasonable, considering the fact that the domains monitored were extracted from online malware and phishing repositories or from spam emails. Most malicious domains are only active for a short period of time before they are discovered and blocked. DIGGER would have collected little-to-no valid data for these dead domains, and they would have been filtered out. Not all hyperlinks in spam belong to malicious or phishing websites; some contain links for legitimate companies peddling wares like cheap pharmaceutical, herbal supplements, online pornography, etc. These companies may not be doing any illegal activities, (or doing them discreetly enough not to be caught), allowing them to utilize stable, legitimate servers. Thus, it is not unreasonable for ≈95% of domains to be removed by the rule-based filter.

Continuing to look at Table IV, we see that MAL and FF domains account for 94.27% of the remaining 5,169 test domains. Again, this is in line with our expectations, since we have already removed the most benign of the non-CDN domains. Since the domain list is generated from suspicious sources, it is reasonable that few would be utilizing the extensive CDN infrastructure typically employed by more popular and reputable domains. Of the 4,873 nefarious domains, ≈96% were MAL domains, with only 179 being FF domains. This result is not surprising, since MAL domains—due to their ease of management—are the traditional and most popular mechanism employed by malicious websites. A MAL domain typically makes use of valid servers rented from less-than-reputable hosting providers. When the domain is discovered and its IPs are blocked, the owner must find a new, shady hosting

provider willing to host the malicious content.

The additional level of misdirection and the nearly limitless supply of IPs enable botnets to make FF domains appealing, despite their more diligent maintenance requirements. Thus far, it has been primarily FFx1_Arec domains observed in the wild, and their popularity is supported with our findings: ≈49% of the FF domains are FFx1_Arec. Unsurprisingly, FFx1_Arec domains are the most popular, since they provide the greatest return on their investment, affording botmasters an additional layer of misdirection without the hassle of maintaining volatile botnet NSes. Botmaster must still monitor the domain and replace the botnet IPs to avoid an interruption of service, but this task is greatly simplified with the use of stable NSes. Unfortunately for botmasters, security professionals have become aware of the FFx1_Arec botnet technique, devising clever detection strategies. While the botnet provides a steady source of fresh A rec IPs, the NSes can still be blocked, crippling the botmaster's control until new NSes can be acquired. As a means of botmasters overcoming this difficulty, we witnessed considerable presence of FFx2 domains, composing ≈21% of the FF domains. FFx2 domains improve upon FFx1_Arec domains by providing an additional layer of misdirection, further protecting the botmaster. Clearly, FFx2 domains require a more diligent management effort than FFx1_Arec domains; in addition to the A rec, the botmaster must constantly replace IPs for the NA rec as well. However, this extra effort also makes FFx2 domains more difficult to subvert, protecting the NSes against simple countermeasures such as IP blocking. Interestingly, when we analyzed the identified FFx2 domains, we found there was a spectrum in the amount of NA rec fluxiness botmasters were incorporating. Obviously, there were domains that were incredibly fluxy in both record types, as demonstrated by *old-and-girl.com* (Fig. 3). Such FFx2 behavior is essentially what we had envisioned when applying the better-known, fluxy A rec behavior to the NA rec. While it's interesting to observe these aggressive FFx2 domains in the wild, it was the FFx2 domains at the other end of the spectrum that proved more insightful. As an example, recall the more modest FFx2 domain *ehuytyt.cn*, shown in Fig. 18. With over 2,500 unique A rec IPs, *ehuytyt.cn* is extremely considerably more more fluxy in its A rec than its NA rec. Using stable bot IPs from its A rec for roughly a quarter of its NA rec IPs, FFx2 domains like *ehuytyt.cn* benefit from the increased control and stability provided by traditional NSes, while simultaneously enhancing the domain's resilience to subversion—for a minimal increase in management—through the use of botnets.

Another interesting discovery is the apparent popularity of FFx1_NArec domains, accounting for ≈30% of the total FF domains observed. Surprisingly, this is a larger share than the FFx2 domains. It seems that botmasters have become aware of security professionals analyzing domains' A recs for FF behavior. Consequently, they have migrated the fluxy behavior to the NA recs, where it is more likely to remain unnoticed. Fig. 19 is a typical example of the FFx1_NArec domains identified by our classifier. It demonstrates a MAL domain strategy for its A rec IPs and a FF strategy for its NA rec IPs. This results in the domain appearing more benign when its A recs are analyzed, while providing the botmaster with a fine level of control over the NSes. Should the domain's malicious activity be detected and the A rec IPs blocked, the botmaster, having retained control over the NSes, can easily replace the IP's with minimal service interruption. The implication of this discovered behavior is straightforward: both record types must be monitored for fluxy behavior in order to quickly identify FF domains and their botnets. A real-time monitor analyzing only domains' A recs will not identify FFx1_NArec domains as fluxy, and it could take days for the A rec's MAL domain behavior to display its slow, steady IP recruitment; even then, the observed recruitment is a side effect of others detecting the malicious domain and blocking its IPs. However, a real-time detection system monitoring NA recs for fluxy behavior could determine the domain to be FF in a much shorter period of time—quite possibly before any MAL domain behavior becomes apparent in the A rec. Obviously, the faster malicious domains can be identified, the sooner they can be shutdown or have their nefarious influence mitigated.

VI. Conclusion and Future Work

In this paper, we examined the global IP-usage patterns exhibited by different types of malicious and benign domains, including FFx1 and FFx2 domains. We have deployed DIGGER, a lightweight DNS probing engine, on 240 PlanetLab nodes spanning 4 continents. Collecting DNS data for over 3.5 months on a plethora of domains, our global vantage point enabled us to identify the various IP-usage patterns inherent to the operation of the different domain types. Conducting a detailed analysis, we were able to determine distinguishing behavioral features between the domain types based on their DNS query results. We have quantified these features and demonstrated their effectiveness for detection by building a proof-of-concept, multi-leveled SVM classifier capable of discriminating between five domain types: CDN, non-CDN, MAL, FFx2, FFx1_Arec and FFx1_NArec. Applying our classifier

on a set of 5,169 unknown domains produced promising results, correctly categorizing the domains with only 2 false positives—due to DNS domain IP parking. Our classification results showed the relative distribution of the domain types in our testing data and the current state of FF domains, including the increased presence and versatile implementation range of FFx2 domains. We have shown that fluxiness is typically more pronounced in A recs, and that there is an apparent trend towards using FFx1_NArec domains, which were previously unseen in the wild.

While our multi-leveled classifier has proven effective in identifying the different domain types, it is only a proof-of-concept detector. It is temporally naive, operating over the complete set of data gathered during DIGGER's ≈3.5-month monitoring period. Additionally, our data was gathered by 240 nodes dispersed around the globe. An optimal and practical detector should function over a much shorter duration, relying on fewer nodes. The problem of determining the optimal monitoring period, the minimal number of nodes, and how to handle anomalous behavior like DNS domain IP parking and node failure remains as future work. Additionally, further study into the cautious MAL domains is required to better detect and subvert them. Lastly, continued analysis and global monitoring of malicious domains by DIGGER should be conducted to keep up with the future direction of malicious domains, improving detection and mitigation strategies.

REFERENCES

[1] Dns-bh - malware domain blocklist. http://www.malwaredomains.com/files/domains.txt, 2009.

[2] Phishtank. http://www.phishtank.com/, 2009.

[3] M. Afergan. Experience with some principles for building an internet-scale reliable system. In *NCA '06: Proceedings of the Fifth IEEE International Symposium on Network Computing and Applications*, page 3, Washington, DC, USA, 2006. IEEE Computer Society.

[4] J. R. Binkley and S. Singh. An algorithm for anomaly-based botnet detection. In *SRUTI'06: Proceedings of the 2nd conference on Steps to Reducing Unwanted Traffic on the Internet*, pages 7–7, Berkeley, CA, USA, 2006. USENIX Association.

[5] F. Boldewint. Peacomm.c - cracking the nutshell. http://www.reconstructer.org/, September 2007.

[6] M. P. Collins, T. J. Shimeall, S. Faber, J. Janies, R. Weaver, M. D. Shon, and J. Kadane. Using uncleanliness to predict future botnet addresses. In *IMC '07: Proceedings of the 7th ACM SIGCOMM conference on Internet measurement*, 2007.

[7] J. Goebel and T. Holz. Rishi: identify bot contaminated hosts by irc nickname evaluation. In *HotBots'07: Proceedings of the first conference on First Workshop on Hot Topics in Understanding Botnets*, pages 8–8, Berkeley, CA, USA, 2007. USENIX Association.

[8] J. B. Grizzard, V. Sharma, C. Nunnery, B. B. Kang, and D. Dagon. Peer-to-peer botnets: overview and case study. In *HotBots'07: Proceedings of the first conference on First Workshop on Hot Topics in Understanding Botnets*, pages 1–1, 2007.

[9] G. Gu, R. Perdisci, J. Zhang, and W. Lee. Botminer: clustering analysis of network traffic for protocol- and structure-independent botnet detection. In *SS'08: Proceedings of the 17th conference on Security symposium*, pages 139–154, 2008.

[10] G. Gu, P. Porras, V. Yegneswaran, M. Fong, and W. Lee. Bothunter: detecting malware infection through ids-driven dialog correlation. In *SS'07: Proceedings of 16th USENIX Security Symposium on USENIX Security Symposium*, pages 1–16, 2007.

[11] G. Gu, J. Zhang, and W. Lee. BotSniffer: Detecting botnet command and control channels in network traffic. In *Proceedings of the 15th Annual Network and Distributed System Security Symposium (NDSS'08)*, February 2008.

[12] B. Guenter. Spam archive. http://untroubled.org/spam/.

[13] T. Holz, C. Gorecki, K. Rieck, and F. C. Freiling. Measuring and detectin fast-flux service networks. In *In Proc. network and Distributed System Security (NDSS) Symposium*, 2008.

[14] T. Holz, M. Steiner, F. Dahl, E. Biersack, and F. Freiling. Measurements and mitigation of peer-to-peer-based botnets: a case study on storm worm. In *LEET'08: Proceedings of the 1st Usenix Workshop on Large-Scale Exploits and Emergent Threats*, pages 1–9, 2008.

[15] J. Hruska. Cracking down on conficker: Kaspersky, opendns join forces. http://arstechnica.com/business/news/2009/02/cracking-down-on-conficker-kaspersky-opendns-join-forces.ars, Feb 2009.

[16] A. Karasaridis, B. Rexroad, and D. Hoeflin. Wide-scale botnet detection and characterization. In *HotBots'07: Proceedings of the first conference on First Workshop on Hot Topics in Understanding Botnets*, pages 7–7, Berkeley, CA, USA, 2007. USENIX Association.

[17] J. Nazario and T. Holz. As the net churns: Fast-flux botnet observations. In *Malicious and Unwanted Software, 2008. MALWARE 2008. 3rd International Conference on*, 2008.

[18] Planetlab. An open platform for developing, deploying and accessing planetary-scale services. http://www.planet-lab.org/, 2009.

[19] M. A. Rajab, J. Zarfoss, F. Monrose, and A. Terzis. A multifaceted approach to understanding the botnet phenomenon. In *IMC '06: Proceedings of the 6th ACM SIGCOMM conference on Internet measurement*, pages 41–52, New York, NY, USA, 2006. ACM.

[20] M. A. Rajab, J. Zarfoss, F. Monrose, and A. Terzis. My botnet is bigger than yours (maybe, better than yours): why size estimates remain challenging. In *HotBots'07: Proceedings of the first conference on First Workshop on Hot Topics in Understanding Botnets*, pages 5–5, Berkeley, CA, USA, 2007. USENIX Association.

[21] A. Ramachandran, N. Feamster, and D. Dagon. Revealing botnet membership using dnsbl counter-intelligence. In *SRUTI'06: Proceedings of the 2nd conference on Steps to Reducing Unwanted Traffic on the Internet*, pages 8–8, 2006.

[22] I. Security and stability Advisory Committee (SSAC). Sac 025 ssac advisory on fast flux hosting and dns. 2008.

[23] B. Stone-Gross, M. Cova, L. Cavallaro, B. Gilbert, M. Szydlowski, R. Kemmerer, C. Kruegel, and G. Vigna. Your botnet is my botnet: Analysis of a botnet takeover. Technical report, April 2009.

[24] P. Wang, S. Sparks, and C. C. Zou. An advanced hybrid peer-to-peer botnet. In *HotBots'07: Proceedings of the first conference on First Workshop on Hot Topics in Understanding Botnets*, 2007.

[25] webhosting.info. Ip to country database. http://ip-to-country.webhosting.info/downloads/ip-to-country.csv.zip, 2009.

[26] L. Zhuang, J. Dunagan, D. R. Simon, H. J. Wang, and J. D. Tygar. Characterizing botnets from email spam records. In *LEET'08: Proceedings of the 1st Usenix Workshop on Large-Scale Exploits and Emergent Threats*, pages 1–9, 2008.