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

Faculty: Kang G. Shin
Graduate Students: Matt Knysz and Xin Hu
The University of Michigan, Ann Arbor, MI 48109-2121

ABSTRACT

Botnet-based hosting or redirection/proxy services provide botmasters with an ideal platform for hosting malicious and illegal content while affording them a high level of misdirection and protection. Because of the unreliable connectivity of the constituent bots (typically compromised home computers), domains built atop botnets require frequent updates to their DNS records, replacing the IPs of offline bots with online ones to prevent a disruption in (malicious) 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 study 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 points 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 have also 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 discover potential trends for botnet-based services, uncovering previously-unseen domains whose name servers *alone* demonstrate fast-flux behavior.

1. INTRODUCTION

A botnet is a vast collection of compromised computers under the control of a botmaster utilizing a 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 illicit profit. 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 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 to send seemingly interesting phishing emails embedded with innocuously disguised 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 host 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 unpredictably 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, they adopt fast-flux (FF) 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 number of IPs per query, botmasters can ensure, with high probability, that the malicious domain resolves to an online bot's IP. An additional level of control and resilience is attained by giving the domain's IP mappings a short time-to-live (TTL) value, allowing botmasters to quickly replace offline bots. Using this FF technique, botmasters effectively turned their botnets into a global Content Delivery Network (CDN), providing highly available and reliable content-hosting services despite frequent node failures/disconnectivity. This extends the lifetime of illegal activities the botnets provide, complicating disruption efforts by introducing an additional layer of misdirection.

Previous research focused on the features of FF botnets and their malicious uses in phishing scams [14] (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

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ORGANIZATION.

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From - To)*
01-06-2011	Final	4/1/2009 - 5/31/2010

4. TITLE AND SUBTITLE
Charlatan's Web: Understanding, Deteeting and Mitigating Botnets

5a. CONTRACT NUMBER

5b. GRANT NUMBER
N000140910753

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)
Kang G. Shin

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Department of Computer Science Engineering
The University of Michigan
2260 Hayward
Ann Arbor, MI 48109-2140

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
ONR REG. Office Chicago-N62880
230 South Dearborn, Room 380
Chicago, IL 60604-1595

10. SPONSOR/MONITOR'S ACRONYM(S)
ONR

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT

Botnet-based hosting or redireetion/proxy serviees provide botmasters with an ideal platform for hosting malicious and illegal eontent while affording them a high level of misdirection and proteetion. Beeause of the unreliable conneetivity of the eonstituent bots (typieally eompromised home computers), domains built atop botnets require frequent updates to their DNS reeords, replaeing the IPs of offline bots with online ones to prevent a disruption in (malicious) service. Consequently, their DNS reeords eontain a large number of constantly-changing (i.e., ``fluxy") IPs, earning them the deseriptive moniker of fast-flux domains---or, when both the content and name servers are fluxy, double fast-flux domains. In this project, we study the global IP-usage patterns exhibited by different types of malieious and benign domains, ineluding single and double fast-flux domains. We have deployed a lightweight DNS-probing engine, called DIGGER, on 240 PlanetLab nodes spanning 4 eontinents. Colleeting DNS data for over 3.5 months on a plethora of domains, our global vantage points enabled us to identify distinguishing behavioral features between them based on their DNS-query results.

15. SUBJECT TERMS
Computer network security, botnets, fast flux, IP usage

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Kang G. Shin
none	none	none	none	14	19b. TELEPHONE NUMBER *(Include area code)* 734-763-0391

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 perspective. In this paper, we attempt to achieve this goal and explore the global usage patterns of botnets' IP addresses. The contribution of our work is four-fold. First, we build a global query engine called *DIGGER* that monitors—for an extended period of time—complete DNS behavior from 240 geographically-dispersed vantage points spanning four continents. This provides us with a global view of how different types of domains vary in their IP-usage patterns. Second, we propose effective methods to characterize and quantify the temporal and spatial IP-usage patterns of FF botnet domains, facilitating the classification and detection of different domain types. This also allows us to reveal several previously-unknown features of FF 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 more than three months' worth of globally-collected data. The results demonstrate the current trend of FF botnets and the effectiveness of using the distinguishing behavioral features we identified with our global DNS-monitoring system.

The remainder of this paper is organized as follows. Section 2 discusses related work. Section 3 defines the terminology we use. Section 4 explores the global DNS IP-usage patterns for different domain types. Section 5 presents our proof-of-concept classifier and its experimental evaluation results. Section 6 discusses the limitations of our system and our future work, and finally, Section 7 concludes the paper.

2. RELATED WORK

Researchers have focused on the operations and threats of botnets by collecting and analyzing bot-related activities, such as IRC traffic [17], spam emails [22], and DNS queries [18]. For example, Rajab *et al.* [17] constructed a distributed infrastructure to measure IRC botnet activities and showed that botnets contribute the majority of unwanted traffic in the Internet. Grizzard *et al.* [6] analyzed the architecture and communication protocol of a most recent P2P botnet, Peacomm (a.k.a. *Storm Worm*) [5], demonstrating that P2P botnets are more robust to node failures and difficult to take down. All these methods fall into the category of passive analysis. To gain a botnet's insider view, researchers also took active approaches, infiltrating botnets with actual malware samples or customized crawlers. For example, Holz *et al.* [12] crafted a specific P2P client to join the Storm Worm's P2P botnet and analyzed its in-depth features. More recently, Stone-Gross *et al.* [20] successfully took over the Torpig botnet for 10 days by preemptively registering DNS domains the bots would be contacting as C&C servers in the near future, allowing them to reveal detailed operations of the botnet and accurately estimate the number of compromised hosts.

Because of the significant threats botnets have posed to Internet services and applications, researchers have proposed various botnet detection approaches. Some exploit the network behavior typical of botnets' C&C protocols. For example, BotHunter [8] attempts to detect bots using IDS-driven dialog correlation based on IRC C&C communication and other common actions taken during the life cycle of a bot. BotSniffer [9] 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, Gu *et al.* proposed BotMiner [7], which clusters similar communication and malicious traffic and performs cross-cluster correlation to identify bot-infected hosts.

Another common method for botnet detection is identifying the unique network patterns of scams they perpetrate. 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 bots, more botmasters have adopted fast-flux DNS techniques to ensure the availability and stability of their malicious service/content. FF techniques were first reported and analyzed as part of the HoneyNet project [21]. Holz *et al.* [11] and Passerini *et al.* [15] both studied the characteristics of FF networks and developed detection algorithms. They gathered URL domains from spam emails and monitored their DNS-query results for an extended period of time, extracting a set of unique features such as number of unique IPs, number of ASes, lifetime of the domains, TTL values, etc. A linear decision function [11] and a naïve Bayesian classifier [15] were applied on these features to identify FF networks. Nazario and Holz [14] later applied a similar approach to track the use of FF domains and characterize additional properties of FF botnets, including their member size, lifetime, and top-level domain distribution. Their results demonstrated that continuous data mining of FF DNS records can yield insights into the operations of FF botnets. More recently, Konte *et al.*[13] studied the dynamics of the FF network from the perspective of online scam hosting infrastructures for different spam campaigns, measuring the change rates of DNS records, distribution of TTL values, and location of IP addresses. Their measurement results suggested that some persistent features may be useful in the detection of FF botnets.

Our work is unique and different from this previous work as follows. First, all of the previous work collected data from a *single* vantage point, and hence, may fail to capture useful features that can only be discovered from a global perspective (e.g., different IP-advertisement strategies used by FF networks, CDN and non-CDN domains). By contrast, we deployed a large number of sensors around the world, providing a global perspective of IP-usage patterns for differ-

ent types of FF botnets (in particular, double FF domains). Second, both approaches in [11] and [15] separate FF domains indiscriminately from normal domains without distinguishing their types (e.g., single and double FF networks). In this paper, we provide detailed categorization of FF domains (including two types of single FF networks and a double FF network) and developed a multi-level classifier capable of discriminating between different types of both FF and non-fluxy domains. This finer-grained classification allows us to gain insight into the current state-of-art and potential trends of different FF botnets, as well as their range in implementation. Moreover, to create an efficient classifier with minimal false positives/negatives, we carefully selected and quantified 7 distinguishing features (some were reported previously [11, 15, 13] but others are new) and apply a subset of features at each level of our classifier. Third, because the purpose of using FF botnets is to reliably distribute malicious content to users despite bot failures, the DNS behavior of FF botnets resembles that of traditional CDNs [4] employing DNS techniques for load-balancing. As a result, some features used in the previous work (e.g., TTL values, IP-change rate, number of unique IP addresses and ASes) appear similar between FF and CDN domains, potentially leading to false positives. In this paper, we conduct a comparative analysis of different IP-management schemes used in FF botnets and popular CDNs. This allows us to accurately filter CDN domains beforehand, and thus, minimize the false positives of the classifier.

3. 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. The primary DNS record components we consider are defined in Table 1.

A rec	• The address (A) record in a DNS query on a domain. • The IP addresses of the domain's content servers.
NS rec	• The name server (NS) record in a DNS query on a domain. • The domain names (not IP addresses) of the domain's NSes.
NA rec	• The A rec in a DNS query on a domain's *name servers*. • The IP addresses of the domain's NSes.
Reverse DNS lookup/name	• The result of a DNS query request for an IP's domain name (i.e., PTR request) • When performing a DNS query on a domain, we also do a reverse DNS lookup on the domain's A and NA rec IPs.

Table 1: *DNS record terminology*

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 our distributed DNS query engine (DIGGER, Section 4.2) was deployed; *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; *IP Index* is a unique index incrementally assigned to each newly-observed IP. In what follows, we explain the terms we use to describe these various domain types and how they behave. Their global behavior will be explained further in Section 4.

FF domains are malicious domains utilizing a fast-flux (FF) DNS-advertisement strategy, typically built atop botnets. Because bots may 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. Domains exhibiting FF behavior in only a *single* record type (i.e., A rec or NA rec, but not both) are considered **FFx1 domains** (single fast flux). More specifically, FFx1 domains that are fluxy in their A recs (i.e., content servers) are termed **FFx1_Arec domains**, while those that are fluxy in their NA recs (i.e., name servers) are termed **FFx1_NArec domains**; FFx1_NArec domains are able to evade current detection strategies that focus on A recs by migrating their fluxy behavior to their NA recs, where it is less likely to be noticed. When FF domains are fluxy in *both* their A and NA recs, they are considered double fast flux, or **FFx2 domains**. 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.

CDN domains are valid, benign domains that uses a CDN, such as Akamai, to improve the delivery of their 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, changing 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 demonstrate geographic awareness (i.e., IPs geographically close to a DNS server will be advertised with higher probability at that server) and load balancing (i.e., techniques improving performance and scalability not observed in FF domains).

Non-CDN domains are valid, benign domains that *don't* use a CDN for delivery of their content. Typically, non-CDN domains use a few stable content servers and a modest number of NSes, with the same A and NA rec IPs appearing in the query results regardless of the queried DNS server's geographic location.

MAL domains are domains that aren't fluxy enough to be considered FF domains, nor benign enough to be considered non-CDN domains. While not necessarily malicious domains, their DNS behavior demonstrates potentially sus-

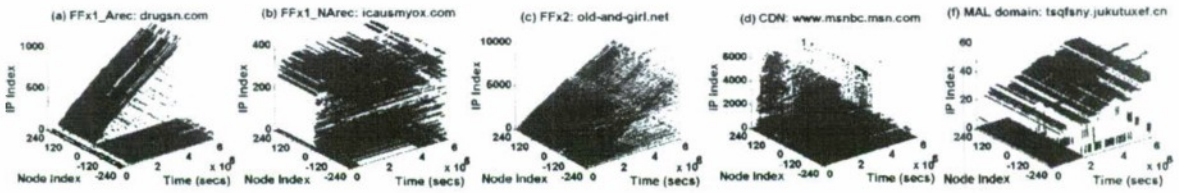


Figure 1: *Global IP-usage (in DNS query results) for some examples of the domain types*

picious behavior often attributed with malicious domains. They tend 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. MAL domains 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 MAL domains may consist of bots or valid servers being used for malicious means. Additionally, benign websites hosted on home computers with dynamic IP addresses could be considered MAL domains by our definition. However, we consider this acceptable since most valid websites are not hosted on home computers, causing those that are to be inherently suspect.

4. GLOBAL IP-USAGE PATTERNS

4.1 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 classification of the different domain types.

4.2 System Architecture

We created a distributed DNS-query engine called DIGGER, deployed on 240 geographically disparate nodes in the PlanetLab testbed [16]. 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.

On each node, for malicious and benign domains, DIGGER performs DNS-query digs on their A rec, NS rec, NA

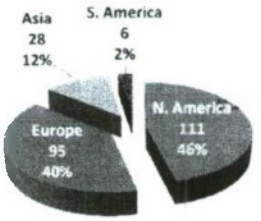


Figure 2: *Global distribution of DIGGER nodes by continent*

rec and the reverse DNS lookup for the A and NA rec IPs. 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 TTL, ensuring fresh DNS-query results. Domains determined to be offline are intermittently dug, so that DIGGER can discover if they come back online. Every 24 hours, DIGGER compresses the raw DNS-query data and uploads the results to our analysis server. This way we aggregate the global DNS-query results for over 106,000 different domains from 240 nodes around the globe. The set of domains monitored by DIGGER is compiled from multiple sources, including online repositories of phishing [3] and malware [1] websites. In addition, we extract domains from URL links embedded in spam emails found in our personal mail boxes, a spam relay trap, and recent additions to online repositories [10] during DIGGER's active period. While DIGGER is active, we continue to gather additionally suspicious domains, adding them to DIGGER's monitoring set. 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 and valid domains, as described in the following subsections.

4.3 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. We discovered 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, resulting in different NSes with identical IPs.

Table 2 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

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

Table 2: *Total A, NA, & overlap IPs for diff domain types*

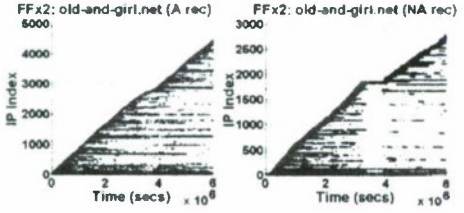


Figure 3: *Global IP usage for example FFx2 domain*

overlap phenomenon was 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. While the representative MAL domains have a small number of total unique IPs (like a non-CDN domain), their IP overlap is exceptionally high, with almost all of their A rec IPs also used for their NA recs, thus setting them apart from valid domains. The IP overlap we empirically observed demonstrates that valid domains use separate machines for their content and name servers—most likely for redundancy and fault-tolerance purposes, preventing a single point of failure. FF and MAL domains, on the other hand, attempt to make the most of their limited resources, reusing IPs for both the A and NA records. Clearly, the amount of observed IP overlap can prove a useful feature for differentiating between valid and malicious domains, especially FFx2 and MAL domains.

4.4 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 confirm/refute this expectation, we have analyzed the advertisement strategies for the various domain

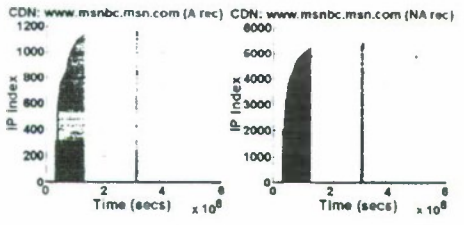


Figure 4: *Global IP usage for example CDN domain*

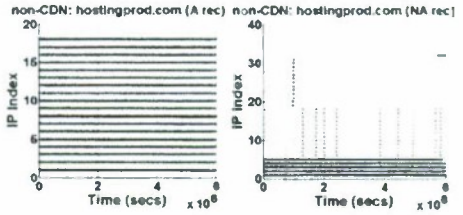


Figure 5: *Global IP usage for example non-CDN domain*

types. For a given domain, we assumed a global vantage point across all DIGGER nodes 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 example FFx2, CDN, non-CDN, and MAL domains in Figs. 3–6, with the y-axis representing the unique IP index and the x-axis representing the time in seconds since DIGGER was deployed. The example domains were added to DIGGER about 1 month after its deployment, resulting in the plots' initial lack of data. 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 curve demonstrates the rate, or speed, with which a domain seems to globally "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, their plots are not included as they are mostly redundant.

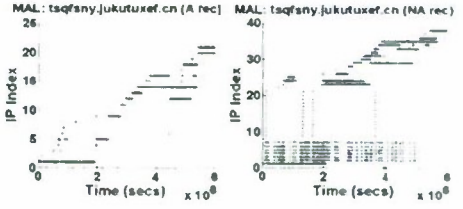


Figure 6: *Global IP usage for example MAL domain*

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.

Fig. 3 shows how a FFx2 domain slowly and nearly continuously accrues unique IPs over its entire online lifetime, with short, intermittent periods of stability. These results indicated that FF domains—consisting primarily of compromised home/office computers that may go offline arbitrarily—must continue to recruit new IPs to help ensure reliable delivery of their nefarious content. In addition, the bots used by FF domains often obtain dynamic IP addresses from their Internet Service Provider (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 IPs.

Meanwhile, when viewed globally, we have discovered that CDN domains (Fig. 4) achieve a much faster recruitment speed, indicating that they advertise IPs from a large pool of stable IP addresses, which they rotate quickly and efficiently for performance purposes, such as load balancing. Our data also reveals that CDNs advertise their IPs in a geographically-conscious manner. For a given CDN domain, a DNS query in Asia will often result in a different set of IPs than would the same query originating in Europe. 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 Europe to European DNS servers. As a result, DIGGER's global perspective observes most of the CDN's IPs in a short period of time. In contrast, FF domains appear to be at the whim of the currently available and online bots, preventing the level of control necessary for geographic IP management. 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 (i.e., nearly the same IPs are seen regardless of query location), whereas a CDN is more localized (i.e., IPs returned are dependent on the queried DNS server's location). Hence, for FF domains, DIGGER's global perspective doesn't allow it to observe many more IPs than at any given local vantage point, resulting in the comparatively slower IP recruitment rate.

From our analysis, we have found that non-CDN domains (Fig. 5) hardly recruit any additional IPs over time. Rather, their IP pools consists of a small number of stable content servers that are almost simultaneously advertised to DNS servers around the world.

Looking at Fig. 6, we can see that *tsqfsny.jukutuxef.cn* demonstrates the slow and somewhat steady recruitment of IPs common to MAL domains. This behavior is likely the result of the MAL domains' malicious activities being detected and their IPs blocked, requiring them to register fresh IPs with DNS in order to maintain content availability. Closer examination reveals that, unlike FF domains which recruit hundreds to thousands of IPs, MAL domains recruit only tens of IPs over more than 3.5 months. This is a drastic difference, and it should prove beneficial in distinguishing MAL domains from non-CDN and FF domains.

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. Our data indicates that non-CDN domains (Fig. 5) use a small pool of very stable IPs with almost no recruitment period; all the IPs used are advertised initially and used throughout the lifetime of the domain. On the other hand, we have found that CDN domains utilize much larger IP pools, from which IPs are advertised based on geographic location and load balancing. When viewed from a global perspective, the fast recruitment speed of CDN domains causes DIGGER to quickly observe most of their available IPs, resulting in a short recruitment period at the onset of monitoring followed by a longer, stable period consisting mainly of previously-seen IPs. As demonstrated in Fig. 4, we can see that the CDN's recruitment period is smaller than its total online period; after its initial recruitment period, the CDN domain stabilizes and advertises a much smaller set of IPs before a quick advertisement spike followed by another stable period. We have also discovered, as shown in Fig. 3, that the fluxy records for FF domains recruit new IPs for nearly the entire duration of the domains' online period, with only short, intermittent periods of stability. That is, nearly the entire time we observe a FF domain to be online, its fluxy records are recruiting new IPs. This constant IP recruitment is a result of DHCP churn and the unreliable nature of the compromised computers serving as bots. The varying recruitment periods we have discovered for the different domain types should provide a useful metric for distinguishing between them.

4.5 Continental Distribution of IPs

Next, we examine how the various domain types differ in their IP distribution (i.e., where the IPs returned in DNS queries are geographically located). We examine the geographic location of IPs based on continent rather than country, because the close proximity of European countries made a country-based resolution too finely-grained. 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 DNS queries are located in a different continent than the queried DNS server? We restate this, for succinctness, as the *percentage of IPs from the wrong continent*.

Q2: From the perspective of each continent containing queried DNS servers, what percentage of IPs returned are located in each continent? Likewise, for succinctness, we restate this as the *continental IP distribution*.

The answer to Q1 can be seen in Fig. 7 for some representative domains. For each domain, we plotted the percentage of A and NA rec IPs from the wrong continent. From Fig. 7, 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 Fig. 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; likewise, we have omitted plots for a MAL domain (since their distribution is functionally similar to non-CDN domains) and for the NA recs' distribution (since the results are similar to those for the A recs). In Fig. 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 *old-and-girl.com* (Fig. 8) indicates the percentage of A rec IPs located in each continent base on queries to Asian DNS servers.

It is interesting to note in Fig. 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. These results clearly reveal the *location-aware DNS advertisement* employed by CDNs. Their DNS query results often contain a majority of IPs located near the DNS server and the issued query, providing fast, reliable services and quicker content delivery to end users by reducing the data's travel distance. Consequently, CDNs demonstrate a smaller percentage of IPs from the wrong continent and a larger variance in continental IP distribution than other domain types.

From our data, we found that MAL and non-CDN domains operate in a similar manner. With a smaller set of stable servers (both content and name) than CDN and FF domains, they don't require complicated load balancing or location-aware DNS advertisement. Instead, we discovered that they adopt a form of *naive DNS advertisement* and indiscriminately advertise their small pool of server IPs around the world nearly simultaneously (Fig. 5). 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. 8 shows that for the non-CDN domain *hostingprod.com*, all of the A rec IPs are in N. America. Because about 46% of our nodes are located in N. America (Fig. 2), we find that 53.77% of *hostingprod.com*'s IPs are from the wrong continent (Fig. 7), approximately (due to rounding) the same percentage as nodes *not* in N. America.

Our analysis suggests that FF domains adopt an advertisement strategy dictated by the unstable nature of their constituent bots, which we term *necessity-based DNS advertisement*. 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 eventually advertise most of their IPs globally. FF domains appear to advertise available IPs to DNS servers around the globe as necessity dictates, with little or no regard to location, resulting in a large percentage of IPs from the wrong continent and a fairly consistent continental IP distribution across continents.

Our findings indicate that the percentage of IPs from the wrong continent and the variance of the continental IP distribution could serve as features for distinguishing CDN domains from the other domain types. This can greatly simplify the detection of FF domains by helping identify them

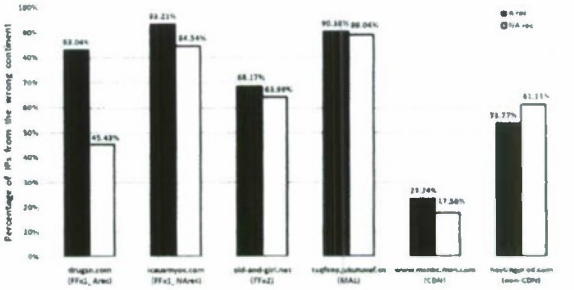


Figure 7: *Percentage of IPs from the wrong continent*

from CDNs, which are often very similar in other respects.

4.6 Number of Unique IP Addresses per Node

Another interesting feature is the number of unique IPs seen across the DIGGER nodes over time. Fig. 9 shows the CDFs for the numbers of unique A and NA rec IPs observed by our 240 DIGGER nodes during the ≈3.5 month monitoring period. Again, MAL domains have been omitted in the plots due to their similarity to non-CDN domains. Our empirical data reveals that non-CDN and FFx1_NArec domains (whose A recs behave like a non-CDN) use a small set of stable content servers. For example, in Fig. 9-(a), neither of them contains more than 18 unique A rec IPs per node. CDN domains are found to exhibit a large number of unique A rec IPs on some nodes, though the number of nodes is considerably fewer than observed for FF domains. For example, for the CDN in Fig. 9-(a), ≈84% of the DIGGER nodes observed less than 22 unique A rec IPs and no node observed more than 200. On the other hand, for the FFx1_Arec and FFx2 domains, we observed a greater number of unique A rec IPs on a larger percentage of nodes. For the FFx1_Arec domain in Fig. 9-(a), ≈45% of the 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, ≈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 number of unique A rec IPs per node appears a promising distinguishing feature, our data implies that this is not the case for the average number of unique *NA rec* IPs. For example, from Fig. 9-(b) it is apparent that CDN and FFx2 domains possess many more unique NA rec IPs per node than the other domain types. Although the CDN domain appears to utilize more unique NA rec IPs per node on average, the FFx2 domain demonstrates a greater number of unique IPs 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 might arise from the CDN trying to ensure the availability of its NSes, afford-

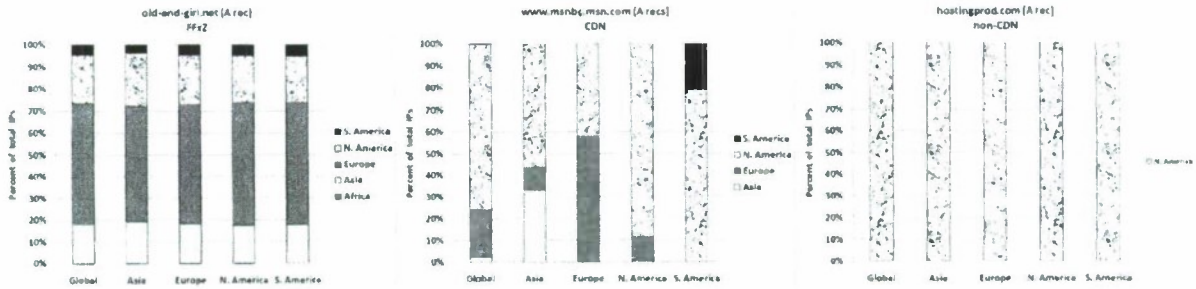


Figure 8: *Percentage of total A rec IPs seen from each continent by DIGGER nodes globally and in each continent*

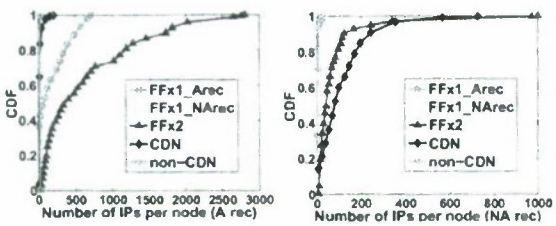


Figure 9: *CDF: # of unique A and NA rec IPs per node*

ing it better control when performing load balancing. In any case, we found the behavior of the FF and CDN domains to be too similar, causing the number of unique NA rec IPs to be an indistinctive feature.

4.7 Total Number of Unique IPs

Based on our data, we learned that non-CDN and MAL domains advertise only a few stable content and name servers with DNS. In the case of non-CDN domains, nearly all the A and NA rec IPs are advertised ubiquitously around the globe. For MAL domains, a smaller number of IPs are used at any given time; however, over time, as the MAL domain becomes detected and its IPs blocked, the number of unique IPs will slowly increase. In either case, the total number of unique A and NA rec IPs for MAL and non-CDN domains is meager when compared to CDN and FF domains (Table 2), making it a useful distinguishing feature.

4.8 Reverse DNS Lookup and TTL

Because an IP's reverse DNS name is set by its ISP and not the owner of the domain, it cannot be faked by a botmaster. This makes it a fairly useful metric for identifying bots, which would contain reverse DNS names typical to home computers (i.e., containing words like comcast, dynamic, dial-up, etc.). Unfortunately, the reverse DNS lookup is highly unreliable, often not returning a result. 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. Therefore, we have decided not to incorporate the reverse DNS name for automatic domain classification, hoping to gain better insight into more reliable features. Instead, when present, we use it to help reinforce or confirm our manual identification of the different domain types.

The A and NA recs' TTL values also appear highly useful for differentiating between the domain types. However, unlike many of the other features we have previously explored, the TTL value is not an uncontrollable consequence of FF domains; it is set by the owner of the domain, affording botmasters a convenient mechanism for circumventing TTL-dependent detection metrics. In addition, it has been shown in [13] that the TTL distribution of FF domains and many popular websites (especially CDN domains) are similar. Consequently, we have decided not to use the TTL as a classification feature. 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.

4.9 Other Features

For the various domain types, we also examined the *average number of nodes per IP* and the *average IP online time*. While these results proved interesting and potentially useful as distinguishing features, we found them less effective than the other features when designing our classifier (Section 5). Therefore, due to space constraints, we have omitted our results concerning these features and refer the interested reader to our technical report for their discussion.

5. DETECTION METHODOLOGY

5.1 Overview

Our observations in Section 4 indicate that the different domain types could be identified based on behavioral features of their global DNS activity. To demonstrate this, we built a proof-of-concept classifier, utilizing a multi-leveled linear SVM (Support Vector Machine). The rest of this section describes its design and implementation, 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.

5.2 Classification Features

Table 3 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 F2, each feature can be applied to a domain's A or NA rec, and while not displayed in Table 3, the features can also be applied to the combined IP pool of the A and NA recs, represented as (A + NA). Lastly, the column labeled "Domain Type Classification Groups" in Table 3 shows how each feature—when applied to the A or NA rec—will likely group the different domain types, represented by square brackets. Table 3 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 3 is a helpful visual tool for determining the application of features at different SVM levels.

Classification Feature	DNS Record	Domain Type Classification Groups
F1. Avg. # of unique IPs per Node	A	• [CDN$_1$, non-CDN$_2$, MAL$_3$, FFx1_NArec$_4$] • [FFx2$_1$, FFx1_Arec$_2$]
	NA	• [CDN$_1$, FFx2$_2$] • [FFx1_NArec$_4$] • [non-CDN$_2$, MAL$_3$, FFx1_Arec$_3$]
F2. A & NA rec overlap	A & NA	• [CDN$_1$, non-CDN$_2$] • [MAL$_3$, FFx2$_2$, FFx1_Arec$_2$, FFx1_NArec$_4$]
F3. % IPs from wrong continent	A NA	• [CDN$_1$] • [non-CDN$_2$, MAL$_3$, FFx2$_4$, FFx1_Arec$_5$, FFx1_NArec$_6$]
F4. Continental IP distribution's average cosine similarity	A NA	
F5. IP recruiting speed	A	• [CDN$_1$] • [non-CDN$_2$, FFx1_NArec$_3$] • [MAL$_3$] • [FFx2$_4$, FFx1_Arec$_4$]
	NA	• [CDN$_1$] • [non-CDN$_2$, FFx1_Arec$_2$] • [MAL$_3$] • [FFx2$_4$, FFx1_NArec$_5$]
F6. IP recruiting period	A	• [CDN$_1$] • [non-CDN$_2$, FFx1_NArec$_3$] • [MAL$_3$] • [FFx2$_4$, FFx1_Arec$_4$]
	NA	• [CDN$_1$] • [non-CDN$_2$, FFx1_Arec$_2$] • [MAL$_3$] • [FFx2$_4$, FFx1_NArec$_5$]
F7. Total unique IPs	A	• [CDN$_1$, FFx2$_2$, FFx1_Arec$_3$] • [non-CDN$_2$, MAL$_3$, FFx1_NArec$_4$]
	NA	• [CDN$_1$, FFx2$_2$, FFx1_NArec$_3$] • [non-CDN$_2$, MAL$_3$, FFx1_Arec$_4$]

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

Table 3: *Features to classify domain types into diff groups*

5.2.1 Feature Quantification

All of the features, except F2, 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 5.3.2. 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, F1 is computed as:

$$\text{F1} = \frac{\sum_{i=1}^{N} P_i}{N} \quad (1)$$

F2: 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, F2 is calculated as:

$$\text{F2} = \frac{|P_A \bigcap P_{NA}|}{\min\{|P_A|, |P_{NA}|\}} \quad (2)$$

F3: Using an online database [2] and *whois* lookup, we were able to determine the country of origin for most IPs observed by DIGGER and the 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 and N be defined as for F1. Then, F3 is computed as:

$$\text{F3} = \frac{\sum_{i=1}^{N} W_i}{\sum_{i=1}^{N} P_i} \quad (3)$$

F4: Let the continents N. America, S. America, Europe, Asia, Africa, Oceania, Antarctic and "unknown" be represented by the numbers 1–8, respectively. Then, n_k = number of nodes on continent k, for $1 \leq k \leq 4$ (continents with DIGGER nodes). For node j on continent k, let $\hat{a}^j$ be a vector representing the number of unique IPs seen from each continent. Thus, $\hat{a}^j_k$ is the number of unique IPs from continent k that were seen on node j. Then, for each continent k with DIGGER nodes, where $1 \leq k \leq 4$, we calculate $\hat{A}^k$ as shown in Eq. (4). We calculate the cosine similarity—shown in Eq. (5)—between every possible pair of vectors $\hat{A}^k$, for $1 \leq k \leq 4$, and then take the average, producing the IP continental distribution's average cosine similarity (F4). 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.

$$\hat{A}^k = \sum_{j=1}^{n_k} \hat{a}^j \quad (4) \qquad \text{Similarity}(\hat{X}, \hat{Y}) = \frac{\hat{X} \bullet \hat{Y}}{\| \hat{X} \| \| \hat{Y} \|} \quad (5)$$

F5/F6: First, we calculate a domain's online time, denoted as T_o. 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 in DNS queries on the domain. If the difference in time between two consecutive *online points* is less than a threshold of several hours, it's added to 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, it's added to T_r. Let P = the total number of unique IPs observed globally for a domain. Then, the IP recruiting speed (F5) and period (F6) are calculated as:

$$\text{F5} = \frac{P}{T_r} \quad (6) \qquad \text{F6} = \frac{T_r}{T_o}. \quad (7)$$

In those instances where all of a domain's IPs are observed instantaneously, resulting in a $T_r = 0$, we set F5 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.

F7: We look at every DNS query gathered by all the DIGGER nodes, counting the number of unique IPs (F7). It represents the total unique IPs used globally by a domain.

5.3 SVM Classifier

5.3.1 Rule-based Filter

Before testing our SVM classifiers, we applied a simple, rule-based filter to remove any domains that were highly unlikely to be CDN, FF or MAL domains. The filter removed a domain from the test set only if ***all*** of the following four rules apply: (1) *none* of its IPs (in both A and NA recs) have a max TTL $<$ 1 day, (2) its A *and* NA recs contain $<$ 10 IPs over the entire monitoring period, (3) *none* of its IPs had reverse DNS lookups containing a suspicious word (e.g., dynamic, dialup, etc.) and (4) *none* of its IPs had reverse DNS lookups indicating it was a known CDN domain (i.e., containing words like akamai).

Both CDN and FF domains would be impractical if none of the IPs had TTL values $<$ 1 day. Similarly, CDN and FF domains should accrue more than 10 IPs after ≈3.5 months of monitoring, as should any MAL domains using stable servers and acting sufficiently suspicious (i.e., their IPs are becoming blocked). Therefore, domains satisfying the first two conditions are extremely unlikely to be CDN, FF or MAL domains. As an additional measure, the filter also ensures that none of their reverse DNS lookups return any suspicious words or indicate that they belong to a CDN. Notice that any domains which were dead (i.e., no IPs) for the entire monitoring period will satisfy all 4 conditions and be removed from the test set. When applied, the filter removed 100,889 unsuspicious or dead domains from our initial set of 106,311 domains, reducing it to 5,422. Finally, we removed 253 domains with insufficient DNS query data (250 domains were momentarily observed by single nodes and 3 domains were monitored by less than 25% of our DIGGER nodes), bringing the test set to 5,169 domains.

5.3.2 Multi-level SVM

Fig. 10 shows the design of our multi-leveled SVM classifier and the results of our training and test sets. Each level of the SVM classifies a domain type from the test set, progressively reducing the number of unknown domains and thus simplifying subsequent classification. Each oval in the figure represents a classified domain type, while each rectangle represents the remaining unknown. 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 test set. We manually identified about 10 representative domains of each type to be used in training, as show in Fig. 10. More difficult to detect by hand, we were only able to manually identify one FFx1_NArec domain.

Table 4 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 each 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. Additionally, the magnitude of *Result* signifies the confidence in classification choice.

As each domain type is classified, it's removed from the set of unknown domains before applying the next SVM level. 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: From Table 3, we see that F3 and F4 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, 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 F3 and F4 on the domains' A and NA recs. As we can see from Table 4, a large percentage of IPs from the wrong continent (F3) or similar IP distributions on each continent (F4) will generate a negative *Result*. We ran SVM-1 on our test set of 5,169 domains. It identified a total of 17 CDN domains, which we manually verified then removed from the test set.

SVM-2: While non-CDN domains advertise all their 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 F5, F6, and F2 on the combined (A + NA) recs, accounting for FFx1 domains demonstrating fluxy behavior in only a single record type. 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 border cases with *Results* closest to 0 and found them to be satisfactorily classified; these results will be discussed further in Section 5.4.1. From Table 4, we can see that F6 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 useful.

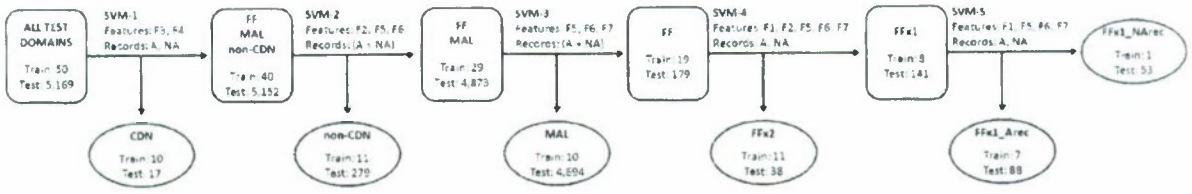


Figure 10: *SVM flowchart*

	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*

Table 4: *Linear SVM equations*

SVM-3: After removing the non-CDN domains identified by SVM-2, the test set was entirely composed of malicious domains (i.e., FF and MAL). Due to the many similarities between FFx1 and FFx2 domains, it's logical to classify MAL domains next. F7 is the most obvious distinguishing feature between MAL and FF domains, but we suspected that F5 and F6 might also prove useful, since FF domains should recruit more IPs over a greater percentage of their online time. SVM-3 applies F5, F6 and F7 to the domains' (A + NA) recs, accounting for FFx1 domains. We trained SVM-3 on a representative set of 10 MAL domains and 19 FF domains. When applied to the test set of 4,873 malicious domains, it identified 4,694 MAL domains and 179 FF domains. Looking at SVM-3 in Table 4, we see that the dominant distinguishing feature is F7: the number of unique IPs. Because of their faster IP recruitment rate, FF domains will quickly outpace MAL domains, resulting in a much larger number of unique IPs.

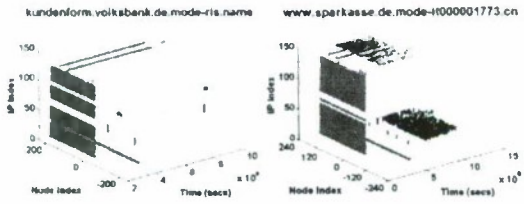
SVM-4: After three stages of the classifier, only FF domains remained in the test set. By definition, the only thing distinguishing FF domains is which record type demonstrates fluxiness. A combination of the two FFx1 domain types, FFx2 domains are the next candidate for classification. From Table 3, it appears that applying F1, F2, F5, F6 and F7 to the individual A and NA recs should discern FFx2 from FFx1 domains. For F1, F5, F6 and F7, all the FF domains will demonstrate fluxy behavior, but the FFx2 domain will demonstrate twice as much as either type of FFx1 domain. Additionally, the IP overlap (F2) experienced by FFx2 domains should be considerably larger. We trained SVM-4 on a representative set of 11 FFx2 domains and 8 FFx1 domains. While F5 appears less significant, F2, F6, and F7 contribute nearly equally in classification, and F1 is a strong indicator of FFx2 domains. These results and their implications are covered in Section 5.4.3. Applying SVM-4 to the 179 remaining FF domains identified 38 FFx2 and 141 FFx1 domains, which we manually verified.

SVM-5: The final level of the classifier is responsible for discriminating between FFx1_Arec and FFx1_NArec domains. With the exception of F2, SVM-5 makes use of the same features and record types as SVM-4 for similar reasons. F2 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. 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 5.4.4. Table 4 shows that F5 and F6 became negligible for SVM-5. F1 holds some influence in classification, but the dominating feature is clearly F7, since the fluxy record type naturally accrues more IPs with time.

5.4 Results

5.4.1 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. 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 Fig. 11 as two long, constant lines with positive Node Index values, indicating parking in the A rec. Appearing as consistent, stable IP addresses, these parked IPs cause a domain to seem 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, it affected classification, account-



(a) (b)

Figure 11: *FFx1_Arec domains with IP parking*

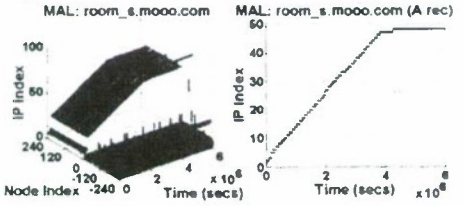


Figure 12: *Cautious MAL domain*

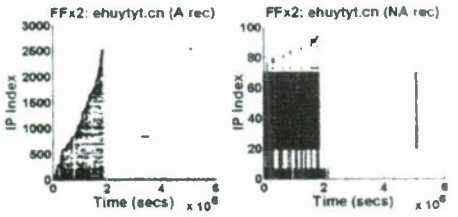


Figure 13: *Classified FFx2 domain*

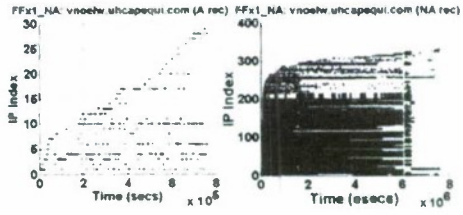


Figure 14: *Classified FFx1_NArec domain*

ing for SVM-2's two false positives. For example, consider the similar domains in Fig. 11. In both cases, the two known nodes were ignored, causing the domain in Fig. 11-(a) to be classified correctly. However, though initially the domain in Fig. 11-(b) appears fluxy, the parking behavior of a large majority of nodes dominates over its lifetime, causing it to be classified as non-CDN. While considered a false positive, this labeling is rather subjective; for the majority of the domain's lifetime it *does* resemble a non-CDN due to IP parking. Since our classifier is temporally naive, using all available data over the ≈3.5 month monitoring period, this misclassification is entirely reasonable.

5.4.2 *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. 12. 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 about once a day, adhering to a meticulously precise schedule. Additionally, we can see from Fig. 12, 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—although 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 the detection and subversion of these domains.

5.4.3 *FF domains*

Another interesting aspect of our classifier is how it distinguishes between the various FF domains. Recall from Table 4 that F1 is the dominant feature for SVM-4, with the NA rec being 4x as influential as the A rec. From Table 2 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 present in the NA rec. The consequence of this asymmetric weighting of fluxiness can be witnessed for the domains in Fig. 13 (classified as FFx2) and Fig. 14 (classified as FFx1_NArec). Both domains demonstrate definite fluxy behavior in one of their record types: Fig. 13 is clearly fluxy in the A rec, while Fig. 14 is clearly fluxy in the NA rec. With only about 20 A rec IPs, the recruitment behavior in Fig. 14 resembles that of a MAL domain, with an IP overlap of less than 4%. Thus, the classifier has 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 Fig. 13 is considered fluxy in its NA rec, which appears relatively stable. Further analysis revealed that the domain has an IP overlap of ≈26%, corresponding with the ≈20–30 NA rec IPs demonstrating fluxy behavior. Since the fluxy A rec contributes over $\frac{1}{4}$ of the NA rec IPs, the less stringent fluxiness demands for the NA rec are met, and the domain is correctly classified as FFx2.

5.4.4 *Domain Type Distribution*

Table 5 shows the number and distribution for each domain type identified by our classifier. Of the 106,311 domains we monitored, our rule-based filter (Section 5.3.1)

identified 101,142 domains as benign or lacking in sufficient data—corresponding to 95.14% of our monitored domains. This is reasonable, considering the 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. Also, not all hyperlinks in spam belong to malicious or phishing websites; some contain legitimate links.

After filtering, MAL and FF domains account for 94.27% of the remaining 5,169 test domains. Since the domain list is generated from suspicious sources, it's unlikely that any would utilize the extensive CDN infrastructure typically employed by popular and reputable domains. Of the 4,873 nefarious domains, ≈96% were MAL domains and 179 were FF domains. This plethora of MAL domains results from their ease of management as the traditional and most popular mechanism employed by malicious websites.

The additional level of misdirection and the nearly limitless supply of IPs provided by botnets 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. Unfortunately for botmasters, security professionals have become aware of the FFx1_Arec botnet technique, devising clever detection strategies. While botnets provide 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. In an apparent attempt by botmasters to overcome this limitation, we witnessed a considerable presence of FFx2 domains, composing ≈21% of the FF domains. FFx2 domains provide further misdirection and protection for the botmaster, guarding the NSes against simple countermeasures at the expense of a more diligent management effort. Interestingly, analysis of the identified FFx2 domains revealed a spectrum in the amount of NA rec fluxiness incorporated by botmasters. Obviously, we observed domains that were incredibly fluxy in both record types, as demonstrated by *old-and-girl.com* (Fig. 3). 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* (Fig. 13). With over 2,500 unique A rec IPs, *ehuytyt.cn* is considerably more fluxy in its A rec than its NA rec. By using bot IPs from its A rec for roughly $\frac{1}{4}$ 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

Domain Type	# of domains	% of ALL	% of MAL/FF	% of FF	% of FFx1
ALL	5,169				
CDN	17	0.33%			
non-CDN	279	5.40%			
MAL/FF	**4,873**	**94.27%**			
MAL	4,694	90.81%	96.33%		
FF	**179**	**3.46%**	**3.67%**		
FFx2	38	0.74%	0.78%	21.23%	
FFx1	**141**	**2.73%**	**2.89%**	**78.77%**	
FFx1_Arec	88	1.70%	1.81%	49.16%	62.41%
FFx1_NArec	53	1.03%	1.09%	29.61%	37.59%

Table 5: *Relative distributions of the various domain types*

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 rec, where it is less likely to be noticed. Fig. 14 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 replace the IPs 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 be detected. However, monitoring NA recs for fluxy behavior could identify the FF domain much more rapidly, providing a quicker response time for mitigating countermeasures.

6. LIMITATIONS AND FUTURE WORK

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. Moreover, our data was gathered by 240 nodes dispersed around the globe. While this might be acceptable for a classifier, an optimal and practical real-time detector should function over a much shorter duration, relying on fewer nodes. The problem of determining the optimal monitoring period and the minimal number of nodes is part of our future work. Our classifier can also suffer from certain anomalous behavior, such as IP parking, which, if dominant, can cause the DNS data to appear benign and result in misclassification. This problem can be solved with an adaptive classifier, utilizing incremental training to improve detection in real time.

Another potential limitation of our work results from botnets using rapid, automatic generation of domain names, such as Torpig [20]. FF domains using this type of botnet will automatically change their domain name on a regular basis, making it difficult to form a long-term picture of the

botnet's activities. DIGGER has no way to predict the future domain names these botnets might utilize. Consequently, our picture of their DNS activity will be incomplete; we will only gather DNS data on such domains until their domain name changes. Since DIGGER continuously updates the list of suspicious domains it monitors, we will be able to gather further DNS data for these botnets under different domain names. Statistical clustering methods based on the observed bot IPs could help by associating different FF domains with the same underlying botnets. However, even this solution suffers from inconsistencies introduced by DHCP churn, warranting further research into this problem. Incidentally, rapidly-and-automatically-changing domain names are typically used by bots as a mechanism for contacting their C&C server; their kaleidoscopic nature makes them ill-suited for phishing or malicious-content campaigns, which require more persistent domain names. Since the focus of this paper is on FF botnets perpetrating such scams, we seldom encounter the swiftly-changing domains used for C&C.

7. CONCLUSION

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 IP parking. Our classification results have shown the relative distribution of the domain types in our test 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.

8. REFERENCES

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

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

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

[4] M. Afergan. Experience with some principles for building an internet-scale reliable system. In *NCA '06*.

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

[6] J. B. Grizzard, V. Sharma, C. Nunnery, B. B. Kang, and D. Dagon. Peer-to-peer botnets: overview and case study. In *Proc. of HotBots*, 2007.

[7] G. Gu, R. Perdisci, J. Zhang, and W. Lee. Botminer: clustering analysis of network traffic for protocol- and structure-independent botnet detection. In *Proc. of the 17th conference on Security symposium*, 2008.

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

[9] G. Gu, J. Zhang, and W. Lee. BotSniffer: Detecting botnet command and control channels in network traffic. In *Proceedings of NDSS'08*, February 2008.

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

[11] T. Holz, C. Gorecki, K. Rieck, and F. C. Freiling. Measuring and detectin fast-flux service networks. In *Proc. of NDSS*, 2008.

[12] 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 *Proc. of LEET'08*, 2008.

[13] M. Konte, N. Feamster, and J. Jung. Dynamics of online scam hosting infrastructure. In *the tenth Passive and Active Measurement conference*, 2009.

[14] J. Nazario and T. Holz. As the net churns: Fast-flux botnet observations. In *Malicious and Unwanted Software*, 2008.

[15] E. Passerini, R. Paleari, L. Martignoni, and D. Bruschi. Fluxor: detecting and monitoring fast-flux service networks. In *Proc. of DIMVA*, 2008.

[16] Planetlab. http://www.planet-lab.org/.

[17] M. A. Rajab, J. Zarfoss, F. Monrose, and A. Terzis. A multifaceted approach to understanding the botnet phenomenon. In *Proc. of IMC*, 2006.

[18] 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 *Proc. of HotBot*, 2007.

[19] A. Ramachandran, N. Feamster, and D. Dagon. Revealing botnet membership using dnsbl counter-intelligence. In *Proc. of SRUTI'06*, 2006.

[20] 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. In *Proc. of the ACM CCS*, 2009.

[21] The Honeynet Project & Research Alliance. Know your enemy: Fast-flux service networks. 2007.

[22] L. Zhuang, J. Dunagan, D. R. Simon, H. J. Wang, and J. D. Tygar. Characterizing botnets from email spam records. In *Proc. of LEET'08*, 2008.