A Lustrum of Malware Network Communication: Evolution and Insights

Chaz Lever[†], Platon Kotzias[*], Davide Balzarotti[∓], Juan Caballero[*], Manos Antonakakis[‡]

{chazlever,manos}@gatech.edu, davide.balzarotti@eurecom.fr,
{platon.kotzias,juan.caballero}@imdea.org

† Georgia Institute of Technology, School of Computer Science,
‡ Georgia Institute of Technology, School of Electrical and Computer Engineering,
∗ IMDEA Software Institute, ∓ EURECOM

***Abstract*—Both the operational and academic security communities have used dynamic analysis sandboxes to execute malware samples for roughly a decade. Network information derived from dynamic analysis is frequently used for threat detection, network policy, and incident response. Despite these common and important use cases, the efficacy of the network detection signal derived from such analysis has yet to be studied in depth. This paper seeks to address this gap by analyzing the network communications of 26.8 million samples that were collected over a period of five years.**

Using several malware and network datasets, our large scale study makes three core contributions. (1) We show that dynamic analysis traces should be carefully curated and provide a rigorous methodology that analysts can use to remove potential noise from such traces. (2) We show that Internet miscreants are increasingly using potentially unwanted programs (PUPs) that rely on a surprisingly stable DNS and IP infrastructure. This indicates that the security community is in need of better protections against such threats, and network policies may provide a solid foundation for such protections. (3) Finally, we see that, for the vast majority of malware samples, network traffic provides the earliest indicator of infection—several weeks and often months before the malware sample is discovered. Therefore, network defenders should rely on automated malware analysis to extract indicators of compromise and not to build early detection systems.

I. Introduction

Malware analysis is at the forefront of the fight against Internet threats. Over the last decade, numerous systems have been proposed to statically and dynamically analyze malicious software and produce detailed behavioral reports [52], [78]. The vast amounts of data collected by such systems can be used to provide important reputation information about both IP and domain name system (DNS) infrastructure, which play an important role in the state-of-the-art detection engines used by the security industry.

Despite the fact that an increasing number of companies and researchers now have access to large malware databases—often containing millions of samples—little is known about how the infrastructure and methods used by Internet miscreants has evolved over time. Previous studies [28], [54], [55], [71], [80], [81] often used small datasets and performed very specific analysis—focusing on topics like the role of cloud providers, the infrastructure behind drive-by downloads, or the domains used by few malware families.

To shed light on this important problem, we report the results of a five year, longitudinal study of dynamic analysis traces collected from multiple (i.e., two commercial and one academic) malware feeds. These feeds contain network information extracted from the execution of more than 26.8 million unique malware samples. We complement this dataset with over five billion DNS queries collected from a large North American internet service provider (ISP). The combination of these two sources provides a unique view into the network infrastructure that malware samples have contacted over the past five years.

Conducting this long term analysis required us to devise a comprehensive filtering process to remove benign domains from our datasets. This process, described in detail in Section III, emphasizes the challenges of reducing the inevitable noise present in any large dataset, and it provides a comprehensive list of steps that an analyst can follow to curate the domains obtained from malware dynamic analysis traces.

Our study also required us to perform the largest malware classification effort to date in order to classify malware samples into families, differentiate malware from Potentially Unwanted Programs (PUP), and correlate domains with their most likely malware families. Our results show that the largest families, by number of samples, are largely dominated by PUP, but traditional malware is responsible for the largest volume of domain resolutions. Our classification enables us to perform the first study comparing the network properties of PUP and malware domains.

After performing rigorous filtering and classification, we proceed with a multi-phase analysis of the dataset. In the first phase, we look at the type, variability, and lifetime of malware domains. Our results suggest that the security community should be cautious in how it uses the results from dynamic malware analysis. For instance, we observed that malware is potentially ineffective as an early-detection trigger as we often observe network activity, in the form of valid resolutions of malware domains, months before the corresponding malware samples are discovered and dynamically analyzed. Specifically, we discovered that 302,953 malware domains were active at least two weeks, in some cases many months, before the corresponding malware samples were analyzed. Thus, organizations that base their network defenses on DNS blacklists and dynamic malware analysis may be unaware of potential threats for significant periods of time.

In the second phase of our analysis, we study the evolution of the IP infrastructure resolved by malware and PUP domains over time, and we identify three interesting categories of "hot spots" in the IP space. These categories correspond to (1) IPs associated with large families that use the same network for extended periods of time, suggesting significant deficiencies in current network and system level defenses; (2) IPs associated with sinkhole operations run by security organizations; and (3) IPs associated with hosting providers that are more willing to tolerate malicious infrastructures, resulting in frequent use by several families. We also analyze the roles of dynamic DNS (DDNS) and content delivery network (CDN) services, as they are both frequently used by malware, and show that approximately 32% of all malware samples in our dataset queried at least one dynamic DNS domain. Finally, we measure the prevalence of domains created by domain generation algorithms (DGAs) in network communication from malware samples, and we find that at least 44% of the domains from dynamic malware traces are generated by 42 DGA families

In summary, our study makes the following core contributions.

First, while dynamic analysis traces can be used as ground truth and forensic evidence of an infection, they should be very carefully curated. We provide (Section III) a detailed and extensive set of rules that network defenders should follow when they wish to remove potentially benign domain names from their dynamic analysis traces.

Second, we observe that PUPs are not only on the rise (Section IV) but also that they surprisingly utilize a very stable network IP infrastructure. Our analysis shows that PUP families host their infrastructure on popular cloud hosting providers and CDNs for up to several years. This may indicate that popular hosting providers do not have the same abuse policies towards banning PUPs that they use to fight malware.

Third, dynamic malware analysis traces are far from the ideal source of information for building early warning systems or detecting new emerging threats. In our analysis, we see that domain names used in malware communications are active weeks, sometimes even months, before malware gets discovered and analyzed by the security community (Section V-B2). This observation has a direct implication on malware domain name blacklists (Section V-A1). While they are certainly useful for detecting current and past malware families, they are not necessarily an efficient method of combating future malware threats. In fact, our long term study shows (Figure 6) that malicious domains were added to major blacklists several days after the malware appeared in one of our feeds and months after the potentially malicious communication was seen in passive DNS.

Beyond these contributions, there are three main differentiators between this work and all the previous work that we build upon. First, we analyze several orders of magnitude more data than any prior research efforts, and we do so over a much longer observation period—almost five full years. This affords us unique insights into how tens of millions of malware samples have evolved over time. Next, we link network level communications (e.g., domains and IPs) with system oriented information (e.g., malware families, PUP). Most existing work does not attempt to perform both these types of analysis in concert—let alone at this scale. Finally, we provide temporal analysis of malware communication over time. This gives us interesting insight into the relationship between the first observable network communication and discovery of malware by the community.

It is only fair to acknowledge that our work was possible because of the many prior efforts in the fields of malware and network analysis—the most notable of which we cite. The fact that our findings confirm the results of many past studies lends further weight to their results and serves to make them more generalizable. We believe the community needs a combination of both large-scale longitudinal studies and more focused small-scale studies. The former better captures global phenomena and general trends while the latter enables more detailed investigations by allowing for manual analysis and deeper inspection of traffic.

Dataset	**Data**	**Count**
Malware Executions	Samples with DNS	26.8 M
	FQDNs	11.5 M
	e2LDs	6.8 M
	IPs	1.4 M
VirusTotal	Reports	23.9 M
Passive DNS	Resource Records	5.2 B
	FQDNs	4.6 B
	e2LDs	2.9 M
	IPs	178.7 M
Public Blacklists	Distinct Blacklists	8
	e2LDs	320 K
Alexa	e2LDs	8 M
Expired Domains	e2LDs	179 M
DGArchive [12]	DGA FQDNs	50 M

TABLE I: Summary of datasets used. All datasets correspond to January 2011–August 2015.

Blacklist	**Target**	**Source**
Abuse.ch	Malware, C&C.	[4]
Malware DL	Malware.	[11]
Blackhole DNS	Malware, Spyware.	[5]
sagadc	Malware, Fraud, SPAM.	[9]
hphosts	Malware, Fraud, Ad tracking.	[7]
SANS	Aggregate list.	[10]
itmate	Malicious Webpages.	[8]
driveby	Drive-by downloads.	[6]

TABLE II: Summary of the public blacklists used in this study.

II. Datasets

Table I summarizes the datasets used in this work. All data corresponds to the time period from January 1st 2011 to August 31st 2015 unless otherwise noted. We use three malware executions datasets to obtain the domains resolved by malware and the IP addresses they resolved to; a passive DNS dataset to map domains to IP addresses and obtain an estimation of their query volume; VirusTotal (VT) reports to obtain additional metadata for the executed malware; public blacklists to identify dates when malicious domains were blocked; the historical Alexa top 1M for whitelisting benign domains; domain expiration dates to mark end of ownership events; and the DGArchive [12] to identify DGA domains. Each of these datasets is described in more detail below.

In this paper, we focus on effective second level domains (e2LDs) rather than fully qualified domains names (FQDNs) because e2LDs better capture domain ownership. For example, the FQDN `www.google.com` has e2LD `google.com`, while `www.amazon.co.uk` has e2LD `amazon.co.uk`, since the second level domain `co.uk` does not correspond to the domain owner. Thus, unless otherwise noted, when we talk about domains we refer to e2LDs and only use FQDNs for better differentiation when needed.

Malware Executions. We collected all the domain names resolved by malware samples from three different datasets—each containing the MD5 of the malware, date of execution in the sandbox, domain names resolved during the execution, and IP addresses that domains resolved to. Each malware sample ran for no more than five minutes in each of the different datasets.

We briefly describe the three datasets but will only refer to their union, after removing duplicate samples, throughout the rest of the paper .

- $UNIVERSITY$. This dataset comes from a university-operated malware execution environment. Collected from January 2011 to August 2015.
- $VENDOR$. This dataset comes from the malware execution environment of a large security vendor that tracks spam and e-mail abuse. Collected from September 2014 to August 2015.
- $ANUBIS$. This dataset comes from the Anubis Web service [42], where users can upload suspicious samples for dynamic analysis. Anubis has operated since 2007, but we focus on executions between January 2011 and June 2014.

In total, we collected the network behavior of 26.8M unique malware samples. It is important to note that this number excludes samples without any valid or successful DNS resolutions.

VirusTotal Reports (VT)**.** VirusTotal [17] is an online service that analyzes files and URLs submitted by users. Submitted executables are scanned with multiple AV engines. VT offers an API to query meta-data on malware samples using a sample's hash, and we queried VT using the 26.8M hashes. For each sample, we collected the time it was first observed by VT, AV analysis date, and AV detection labels. Of the 26.8M samples, 89% were known to VT at the time of our submission (i.e., during the period 2015-16).

Passive DNS ($pDNS$)**.** Due to agreements with the provider of this data, we cannot publicly disclose the exact source, but we can state that this dataset contains passive DNS data collected from a large ISP in the United States. It contains the domain names resolved by clients of the ISP and the IP addresses those domains resolved to. This data was collected above the recursive DNS server, and therefore, it does not contain information about the clients making requests—rather it aggregates resolutions from all clients. In particular, the dataset contains resource records (i.e., timestamp, queried domain name, and associated RDATA [48], [49]), as well as domain lookup volumes aggregated on a daily basis. It comprises 2.9M e2LDs resolving to 178.7M IP addresses.

Public Blacklists (PBL)**.** This dataset contains 320K malicious e2LD entries extracted and aggregated from the eight public domain blacklists, detailed in Table II, which we regularly collected and updated for the entire duration of the project. Due to this aggregation, the dataset includes multiple types of abusive domains such as drive-by downloads, phishing, and botnet C&C. These domains are curated by members of the security community and, thus, represent cases of human verified abuse. For each domain, the data also provides the exact date when the domain was included in the blacklist.

Alexa. This dataset contains rankings of the Alexa top million domains collected daily [2]. It contains approximately 8M unique e2LDs across our entire analysis period.

Expired Domains. This dataset includes the expiration dates of 179M (benign and malicious) e2LDs for the past seven years. These expirations were verified by recording removals from successive gTLD zone transfers and, since the removal alone does not always indicate an expiration, were further vetted using the Extensible Provisioning Protocol (EPP) with a domain reseller account. This methodology is modeled off of previous work that studied potential pitfalls resulting from domain ownership changes [43].

DGArchive. Plohmann et al. [58] recently reverse-engineered malware families that use a DGA. The results of their work is collected in the DGArchive [12], a database of 50M domains that can be generated by the DGAs of 66 malware families. We use the DGArchive to identify DGA domains among the domains resolved in the malware executions.

Limitations and Potential Biases. Despite our best efforts to collect the most comprehensive set of data sources to perform our study, there are still some limitations and potential biases worth mentioning. For example, our study cannot cover samples that have failed to run or that used evasion techniques to avoid revealing their network behavior in the analysis sandbox. To ameliorate this issue, we combine three different malware feeds each using their own sandbox environment. Our datasets also have some geographical bias towards the United States, since the passive DNS data was collected from a large US ISP. However, we believe some form of bias is unavoidable in this type of study. Compared to the state of the art in DNS and malware analysis, our datasets still provide the broadest and deepest view on malware network behavior to date by far.

III. Domain Filtering

We start our analysis by processing the DNS requests performed by malicious files run in dynamic analysis sandboxes. For this, we first remove 255,747 samples that were not flagged as malicious by any AV vendor (according to the results collected in our VirusTotal dataset). What was left was a set of likely malicious or unwanted files.

Since malware does not interact with exclusively malicious infrastructure, not all domains queried by malware samples can be considered malicious. In fact, as it is often the case with large datasets, the initial set of DNS requests was very noisy and needed to be carefully pre-processed to remove all spurious and unwanted entries. In our study, we want to focus on domains that are associated with actual malicious communication. However, despite the fact that all domains are requested by malicious files, the vast majority of the requests in our dataset did not fall in this category. While this may seem surprising at first, it is the consequence of several factors—such as the presence of non-existent domains generated by domain generation algorithms (DGAs), connectivity tests to benign domains, sinkholes, and spam-related activity.

To remove this noise, we proceeded with an initial filtering phase divided into four separate steps with the goal of eliminating invalid, benign, or sinkholed domains as well as reverse delegation queries.

Invalid Domains. Since not all DNS requests result in a valid resolution, we first filter out DNS queries that request non-existent domains (i.e., that do not return a valid IP address). This step is particularly important to reduce the impact of domain generation algorithms (DGAs), where malware tries to resolve many possible domains until it finds one that has been registered by the botmaster. We study the resolutions for non-existent domains, which may be subsequently registered and used for abuse, and DGA behavior in Section VII.

Overall, this first filtering step successfully reduced the number of unique effective second level domains from 6,850,793 to 1,316,331 , and the number of fully qualified domain names from 11,532,653 to 3,767,234 .

Benign Domains. The hardest part of domain filtering consists of identifying and removing the queries performed towards benign domains. Their presence is due to many factors that include malware using legitimate services (e.g., Dropbox), testing if the infected machine has a working Internet connection, downloading components from compromised websites, delivering spam messages to victim mail servers, and even querying an existing benign domain as a result of collisions in a poorly designed DGA algorithm.

The variety of potential causes makes it very difficult to automatically filter out all benign DNS requests. Our approach relies on three separate steps. In the first, we use the $ALEXA$ dataset to remove domains that appeared in the Alexa top ten thousand most popular domains for at least a year, with the exception of dynamic DNS domains—which are often abused for malicious purposes. While the $ALEXA$ dataset provides a good starting point, it fails to capture some obviously popular domains. Therefore, in the second step we manually sifted through the most popular domains remaining after the Alexa filtering, and we identified and removed from our dataset other popular sites such as content distribution networks. This step reduced the set of effective second level domains from 1,316,331 to 1,291,313 and fully qualified domain names from 3,767,234 to 3,295,860.

Finally, we noticed that the remaining dataset was largely dominated by spam bots, which query hundreds or even thousands of benign domains with the goal of locating the SMTP servers of their targets. A comprehensive study of spam behavior is outside the scope of this study. Therefore, we used an aggressive filter that removed any samples performing MX lookups and, as some malware may receive a pre-generated list of MX records, samples that queried for domains containing mail-related keywords (e.g., mail, smtp, imap). While excluding entire samples matching this filter may seem aggressive, we observed that only 405,742 (1.5%) distinct samples contained at least one MX or mail related domain. The presence of these domains suggests a different type of behavior from the rest of the samples in our dataset, and therefore, we chose to discard them to avoid missing less popular, benign domains they may have queried.

In total, their removal reduced the set of effective second level domains from 1,291,313 to 329,348 and fully qualified domain names from 3,295,860 to 2,154,609.

Reverse Delegation Zones. DNS Pointer Records (PTR) often reflect activity from system processes (e.g., gethostbyname()) trying to resolve IP addresses in a remote network. This can occur when a program directly connects to an IP address without performing a DNS resolution of a service's domain name. For example, Windows logging makes note of a network socket connection but avoids listing the IP address—generating a DNS PTR record instead. This behavior, associated with Windows logging of RFC 1918 [53] host names, can be observed at the root levels of DNS [25]. Thus, dynamic execution of a malware may generate reverse delegation domain names that point to remote residential IP space. While the IP could be malicious, the reverse delegation domain name and its effective second level domain cannot be considered malicious as they are typically owned by the ISP (e.g., Verizon) or the hosting provider (e.g., Rackspace).

While it may seem reasonable to remove all e2LD domains seen in PTR records, this would result in too coarse of a filter because the owner of the netblock has the power to assign any domain as the reverse DNS pointer. Thus, some PTR domains will contain the actual domain name used to resolve an IP address instead of a domain, created by the ISP or hosting provider, to describe the underlying infrastructure.

In our final step, we remove benign PTR domain names from our malware domain dataset by excluding zones used by large ISPs and hosting providers for reverse DNS delegation [21]. In simple terms, reverse DNS is the domain name that an Internet provider has delegated to an IP address. For example, for the IP address `173.53.80.48` the Internet provider has assigned the following reverse DNS delegation: `static-173-53-80-48.rcmdva.fios.verizon.net.` This domain name can be retrieved by asking the PTR DNS record of the original IP.

Since malware execution may result in DNS PTR records to be created, we want to exclude the most frequently witnessed

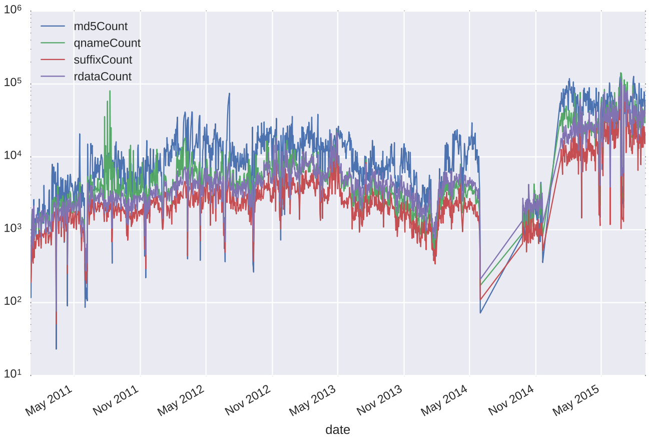


Fig. 1: Number of malware samples, qnames, e2LDs, and IPs according to the execution time of the samples.

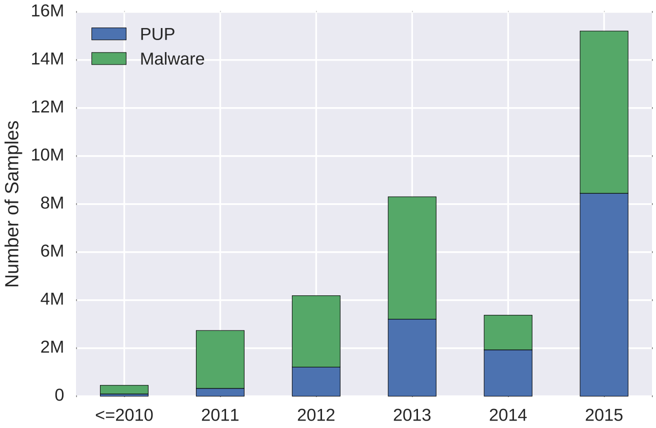


Fig. 2: Malware and PUP samples over time. The drop in 2014 is due to a downtime of our largest feed of malware executions.

e2LDs in such reverse delegation. Therefore, we obtained a PTR scan of of all IPv4 from the Internet Systems Consortium (ISC), and we broke down all the e2LDs in this datasets according to the number of /24 and /16 network that can be seen. Our assumption here is that if the same e2LDs can be seen in several /16 and /24 networks, it must reflect a reverse DNS allocation conducted by an ISP or a hosting provider. We decide to pick the top 1% of the e2LDs for both /16 and /24 networks in our datasets, which reflects e2LDs that have been seen in more than ten /24 and /16 networks at the same time. This methodology identifies only 4,323 e2LDs—resulting in reverse pointer domains that are very likely associated with ISP or hosting networks. We use this list as the final filter, reducing the set of effective second level domains to 327,514 and 2,085,484 fully qualified domains.

Filtering Summary. The domain filtering phase reduced the initial candidate set of domains queried by malicious samples by over 95%. However, despite the significant reduction in e2LDs, 20M malware samples remain after filtering with at least one valid resolution.

Overall, this filtering was a very challenging and time-consuming process and the final result, as we will discuss later in the paper, still likely contains some benign domains with low popularity. However, we believe that our effort emphasizes two very important problems. First, the vast majority of DNS queries performed by malware are not malicious per se – and this may have a large impact on those approaches that populate domain blacklists based on the results of dynamic analysis sandboxes. Second, performing studies on very large datasets requires long periods—months in the case of this work—of manual work to tune filters and properly remove unwanted noise.

The final distribution of samples and domains over the four years of our dataset is summarized in Figure 1. The drop in the second half of 2014 reflects a failure in our collection infrastructure for the largest feed of malware executions.

IV. Classification

We perform 3 classifications on our dataset: grouping samples into families, classifying families as either malware or PUP, and assigning e2LDs to specific families.

Sample classification. We cluster and label all samples into known families using the AV labels in VirusTotal reports. While AV labels are known to be noisy [22], [50], we leverage AVClass [66], a recently released open-source tool for massive malware labeling. AVClass successfully removes noise from AV labels by addressing label normalization, generic token detection, and alias detection. The tool achieves F1 measures between 0.94 and 0.70 and it can process extremely large sets of VT reports—each containing AV scans of one sample by multiple AV engines. AVClass outputs for each sample the most likely family name and a confidence factor based on the agreement across engines.

PUP/Malware family classification. In addition to the family, AVClass also outputs for each sample whether it is PUP or malware by examining PUP-related keywords in the AV labels. However, that classification is conservative as AV vendors often do not flag PUP samples as such. Thus, some samples in a family may be flagged as PUP and other samples in the same family as malware. To address this issue we have modified AVClass to output a classification for each family as PUP or malware, so that all samples in the family can be considered of the same class. Our modification counts for each family the number of samples flagged as malware and PUP. Then, it applies a plurality vote on the samples of a family to determine if the family is PUP or malware. We have contributed this modification to AVClass to be integrated in the tool.

We use our modified AVClass to automatically cluster and label 23.9M samples for which we have a VT report. As a comparison, the previous largest malware clustering/classification effort in the literature was the AVClass evaluation with 8.9 M samples [66]. Figure 2 shows the number of malware and PUP samples over time. The figure shows an increase of PUP samples over time, with PUP overtaking traditional malware since 2014. Kotzias et al. [38] observed the same trend but on a dataset two orders of magnitude smaller and from a single

Rank	Family	Samples	Type	e2LDs	FSeen
1	vobfus	2.8 M	Malware	741	11/09
2	multiplug	2.4 M	PUP	808	01/13
3	loadmoney	1.6 M	PUP	2,958	12/12
4	virut	1.4 M	Malware	40,705	03/08
5	softpulse	1.3 M	PUP	3,793	06/14
6	hotbar	1.1 M	PUP	306	08/10
7	installerex	847 K	PUP	155	12/11
8	firseria	795 K	PUP	3,138	07/12
9	outbrowse	771 K	PUP	52	04/13
10	installcore	661 K	PUP	1,118	09/11
Top 10		49%	-	15%	-

TABLE III: Top 10 malware families by number of samples in our dataset. The FSeen column contains the first seen date of a family by VirusTotal.

Rank	Family	e2LDs	Type	Samples	FSeen
1	virut	40,705	Malware	1.4 M	03/08
2	rodecap	17,382	Malware	11.8 K	05/09
3	zbot	12,959	Malware	163 K	01/08
4	tedroo	6,272	Malware	5 K	11/08
5	sality	4,964	Malware	463 K	12/08
6	upatre	4,658	Malware	503 K	09/13
7	fareit	4,217	Malware	61 K	10/11
8	softpulse	3,793	PUP	1.3 M	06/14
9	ircbot	3,635	Malware	28.5 K	05/06
10	firseria	3,138	PUP	795 K	07/12
Top 10		31%	-	17%	-

TABLE IV: Top 10 malware families by number of filtered e2LDs that resolved to a valid IP address. The FSeen column contains the first seen date of a family by VirusTotal.

source, for which they could not discard source bias. Other work has also hinted on the prevalence of PUP. Thomas et al. [72] showed that ad-injectors affect 5% of unique daily IP addresses accessing Google. They also measured that Google's Safe Browsing generates three times as many detections for PUP as for malware [73].

AVClass identifies 17.7 K non-singleton families. Table III presents the top 10 families, which comprise 49% of the samples and are largely dominated by PUP. The largest malware families are `vobfus` (a Visual Basic worm [47]) and `virut` (a virus that appends its payload to other executable files [70]). Both families have self-replicating behavior that increases their polymorphism. The PUP families include adware that modifies advertisements or searches in the browser (`multiplug`, `loadmoney`, `hotbar`) [72] and a number of pay-per-install (PPI) programs (`softpulse`, `installerex`, `firseria`, `outbrowse`, `installcore`) [37], [73].

The 3,834 families with more than 10 samples comprise over 90% of all samples. Of those families, 3,165 are malware and 669 PUP. While there are more malware families, the PUP families are larger with an average of 16 K samples per family compared to 3.5 K for malware families. This illustrates the highly polymorphic nature of PUP, which is not due to self-replication, but likely due to evasion of AV engines [38].

e2LD classification. We create a mapping from e2LD to the most likely family the e2LD belongs to. For this, we first create a mapping from e2LD to the number of samples of each family that have resolved that e2LD. Then, for e2LDs that have been resolved by at least 10 samples, we assign each e2LD to the family with most samples resolving it. e2LDs with less than 10 samples resolving them are left unclassified.

Table IV presents the top 10 families ranked by the number of resolved e2LDs. Compared to the ranking by samples, this ranking is dominated by malware (8/10 families), which may indicate that PUP families have a more stable domain infrastructure and malware uses higher levels of domain polymorphism. The e2LDs from these 10 families correspond to 31% of all filtered e2LDs and Virut alone is responsible for 12% of them. Virut's domain infrastructure comprises a few dozens stable domains in the `.com` and `.pl` (Poland) TLDs, as well as a DGA. We study DGAs in Section VII. The popular `zbot/zeus` botnet is ranked third in Table IV, likely due to many different operators using the botkit.

By combining the e2LD to family mapping and the family to PUP/malware mapping, we can mark e2LDs as belonging to malware or PUP. This classification identifies 36.5 K malware and 9.1 K PUP e2LDs. The remaining e2LDs are left unclassified due to less than 10 samples resolving them. This classification enables to study separately and compare properties of the PUP and malware network infrastructure (Section V-B).

V. Malware Domain Analysis

Malware often engages in various network communications in an attempt to exfiltrate data, communicate with a command and control (C&C) server, or download additional illicit software. This communication often relies on DNS rather than static IP addresses to provide resiliency against IP blacklisting and ensure an overall agility for the malicious operation.

Therefore, we study the domains queried by malware to better understand the temporal DNS properties of their network communications. In Section V-A, we evaluate domain names queried by malware during dynamic malware analysis. Our experiments show that malware frequently uses domain polymorphism that significantly limits the network policy and detection abilities of DNS blacklists. Then, in Section V-B, we correlate those domain names with a large passive DNS dataset to identify whether we first collect the malware sample or observe passive DNS activity for malware domains on the network. We find that a significant percentage of malware domains can be seen in passive DNS several weeks, in many cases even months, before the actual malware sample was dynamically analyzed by the security community.

A. Dynamic Malware Analysis

We start by analyzing domains collected from dynamic malware analysis. As noted in Section II, we have a dataset of 26,853,732 malware samples collected since January 2011. From these samples, we collected 11,532,653 fully qualified domain names under 6,850,793 distinct effective second level domains. After extensive filtering, detailed in Section III, we reduced this to 2,085,484 fully qualified domain names under 327,514 effective second level domains. In the following sections we study various properties of this set of domains.

1) Domain Polymorphism: Once a sample has been analyzed, the domain names used to facilitate malicious communication can be added to a DNS blacklist. Obviously, the effectiveness of these blacklists depends on how often different malware reuse the same domain names.

The analysis of domains resolved by samples in our dataset shows that most malware samples appear to use different domains over time, as shown in Figure 3. In particular, Figure 3a shows that most MD5s resolve less than 10 unique e2LDs. Even more interesting, most of these e2LDs were seen only a single time across our five year collection period (Figure 3d), which means that they were only queried by a single malware sample. *This is an interesting result because it suggests that most domains are used only once by a single malware sample in our dataset.* If the domain is embedded in the binary and not downloaded from an external source, this can also cause samples in the same family to have different MD5s, even in absence of other polymorphism techniques. Furthermore, Figure 3b suggests that network evasion is being done predominantly on the e2LD since the majority of e2LDs have few child FQDNs. Further reinforcing this result, Figure 3e shows that FQDNs share an almost identical distribution to e2LDs.

These results suggest that *blacklisting malware domains observed during dynamic analysis does little to prevent future communication from newly discovered malware samples*. This result does not diminish the usefulness of collecting malware samples or performing malware analysis, but simply underline the limitation and reactive nature of relying on malware samples DNS queries for threat mitigation.

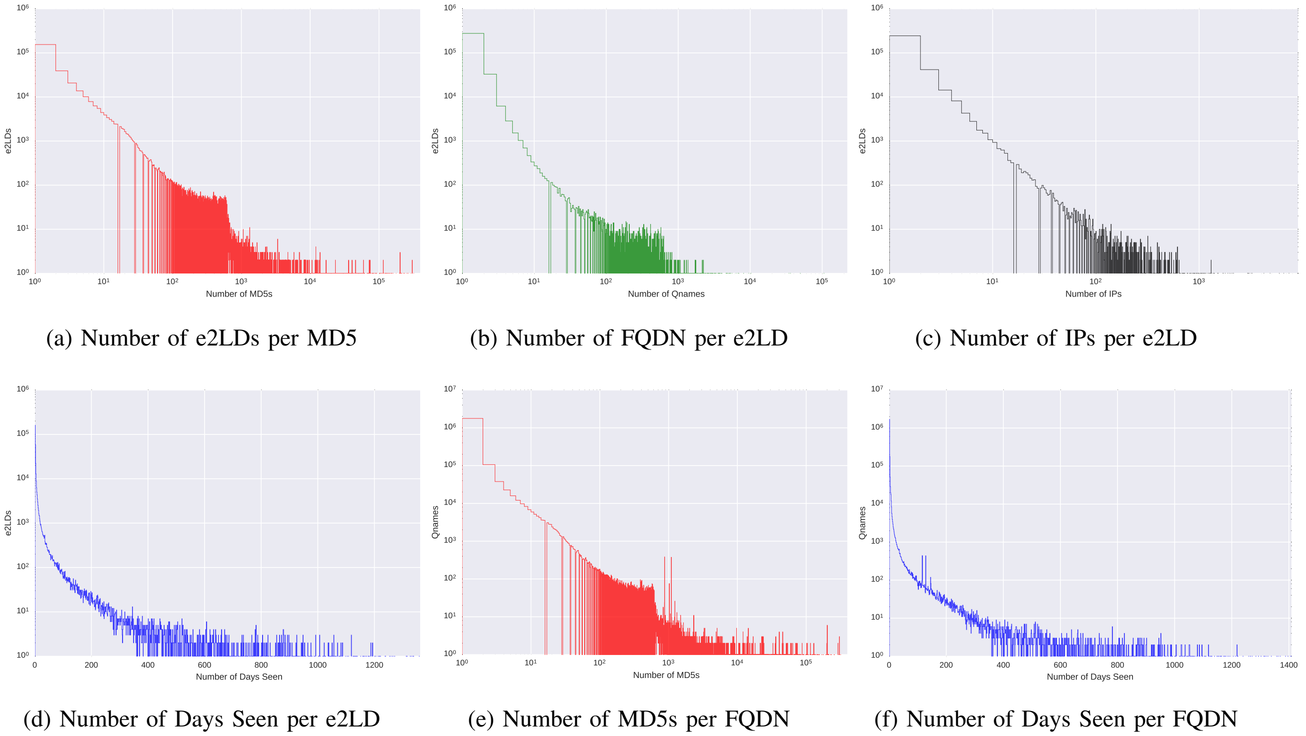
2) Dynamic DNS: Dynamic DNS allows nameservers to be automatically updated with frequently changing information. For example, users with dynamically assigned IP addresses commonly use dynamic DNS as a way of accessing their devices through an easy to remember domain name, which is updated as their IP address changes. There are numerous publicly available services that provide this functionality (e.g., [14], [16]), and many of these services allow users to select a subdomain under a domain owned by the dynamic DNS provider, eliminating the need for the user to register a new domain name.

Due to its ability to provide rapid updates, dynamic DNS is also abused by malware authors to point domains at C&C servers or infected hosts. Furthermore, by using a domain provided by the dynamic DNS provider, the abuse cannot be blocked at the zone level without also blocking other legitimate users of the service. In fact, this has caused significant problems for past remediation efforts [3]. Additionally, Previous work [29] has shown that dynamic DNS domains are blocked at a higher rate (0.2%) than for all other web traffic (0.001%) as measured by data collected from Cisco Cloud Web Security (CWS). This suggests that there is a higher incidence of abuse for dynamic DNS domains. Unfortunately, no information was given about the scope or observation period of the data used to arrive at these numbers. In Section V-A2, we used our dataset spanning five years and 26.8M malware samples to perform a similar analysis. While we arrived at a similar conclusion that malware frequently makes use of dynamic DNS, our list of most frequently used dynamic DNS domains differed substantially this previous work. Since the popular dynamic DNS domains referenced in the previous work were a subset of those used in our study, this may indicate that the popularity of dynamic DNS domains for abuse varies over time.

Therefore, we decided to analyze which dynamic DNS providers are most frequently used by malware. In total, we found 718 known dynamic DNS e2LDs in our dataset from a list of dynamic DNS domains gathered from two public sources [13], [15]. Figure 4 reports the top 100 of these, sorted by the number of malware samples querying them. The most popular dynamic DNS domain, `dnsd.me` (owned by the dynamic DNS provider DNSdynamic [1]), was queried by 216,221 unique MD5s . This service is not only free, but it also offers unlimited registrations and an API for account management—making it very attractive for malware authors. Including `dnsd.me`, the top 50 dynamic DNS domains each have at least a thousand distinct malware samples that query them, and on average each of those domains has approximately 366 subdomains under it. In fact, we see that these top dynamic DNS domains account for 19,766 FQDNs. When looking at all 718 dynamic DNS domains, we see that they are queried by 8,675,449 distinct malware samples, which represents approximately 32% of all malware samples with DNS queries. Furthermore, these 718 domains account for 51,350 FQDNs. Thus, unlike most of the domains we discussed in Section V-A1, *dynamic DNS domains are commonly used across many malware samples and evasion is performed on the child label of the domain.*

3) Content Delivery Networks: Content Delivery Networks (CDNs) are frequently used to serve content from multiple, geographically distributed, data centers to provide increased performance and availability. By taking the client location into account, CDN providers are able to serve up content from the nearest data center, improving network performance. Most providers are still able to offer performance benefits even when location information is unavailable due to faster connections and high-end data centers. Additionally, serving content from multiple data centers helps obviate content outages by providing network redundancy. It is no surprise, given their benefits, that CDNs are widely used on the Internet.

In this section, we study how malware uses CDNs by studying domains collected from dynamic malware analysis. Figure 5 shows a plot of all CDN domains, sorted by how many unique malware samples queried them in our dataset. The first notable feature of this plot is the discrepancy between the most and least popular CDNs. The top five most queried CDN domains include `akamai.net`, `edgesuite.net`, `cloudfront.net`, `netdna-cdn.com`, and `akadns.net`. This list includes some of the largest CDNs and is not dissimilar from what a benign network application might be seen querying. Another interesting insight from Figure 5 is the number of malware samples using CDNs. The `akamai.net` domain alone is queried by 2,183,352 distinct malware samples and has 1,492 unique child labels under it. *The large number of child labels combined with potentially benign usage allows malicious content hosted in a CDN to effectively hide in plain site.*



(a) Number of e2LDs per MD5

(b) Number of FQDN per e2LD

(c) Number of IPs per e2LD

(d) Number of Days Seen per e2LD

(e) Number of MD5s per FQDN

(f) Number of Days Seen per FQDN

Fig. 3: Shows histograms of MD5 network traces broken down by various components.

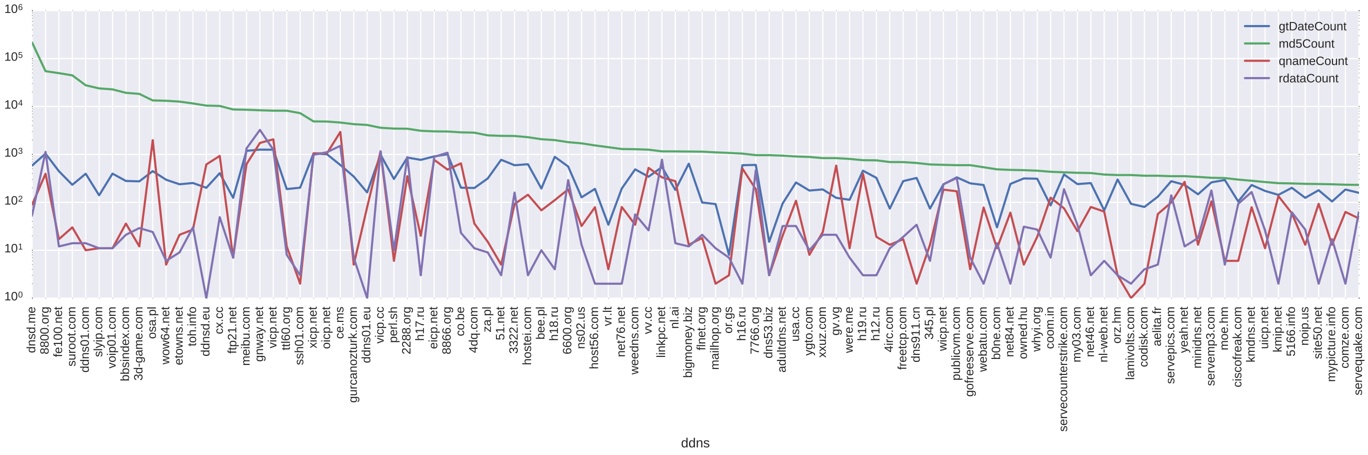


Fig. 4: Top 100 most popular Dynamic DNS domains queried by malware samples.

B. Passive DNS and Blacklists Analysis

In the previous sections, we analyzed the domains collected from network traces observed during dynamic analysis. In this section, we correlate these domains with three other data sources: (1) a passive DNS dataset provided by a large ISP in the United States, (2) a number of public DNS based blacklists, and (3) a set of domain expiration events. This allows us to study the lag between when a domain is discovered through dynamic malware analysis, or listed on a blacklist, and when it is first resolved in passive DNS. This allows us to better understand the implications of relying on dynamic malware analysis of public blacklists for early detection systems.

1) First Appearance: We start our analysis by evaluating efficacy of public blacklists at identifying malware domains. This provides an interesting perspective because domains on these lists have already been flagged as abusive by manual experts or dedicated services. The result of this analysis is plotted in Figure 6a, separated by the type of sample. As we explained in Section IV, we classify domains in our malware analysis traces as belonging to a malware family, PUP, or an unclassified category—which comprises e2LDs resolved by less than 10 samples. This separation allow us to provide

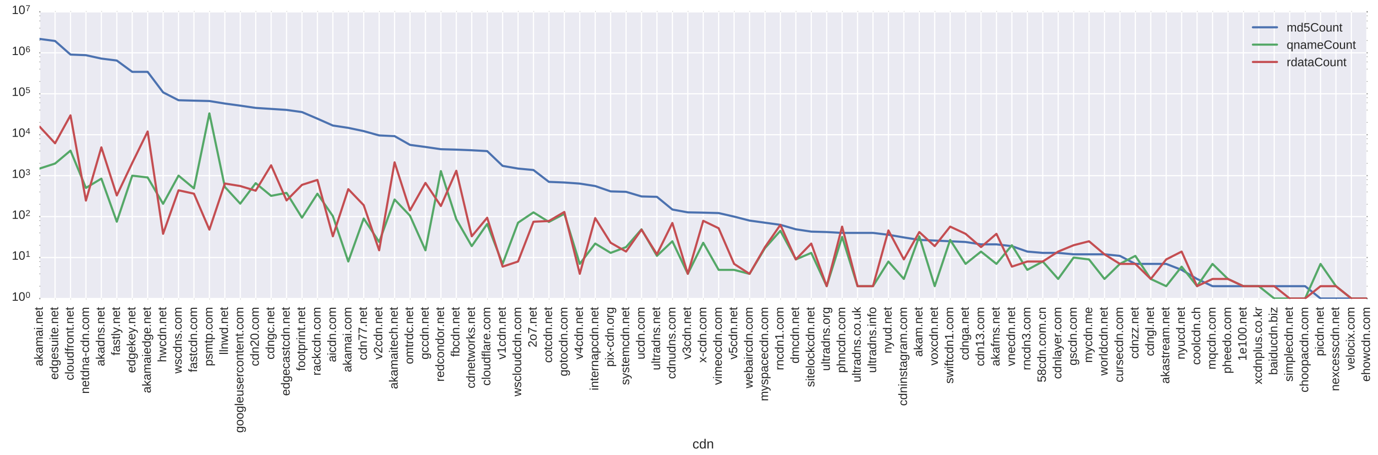
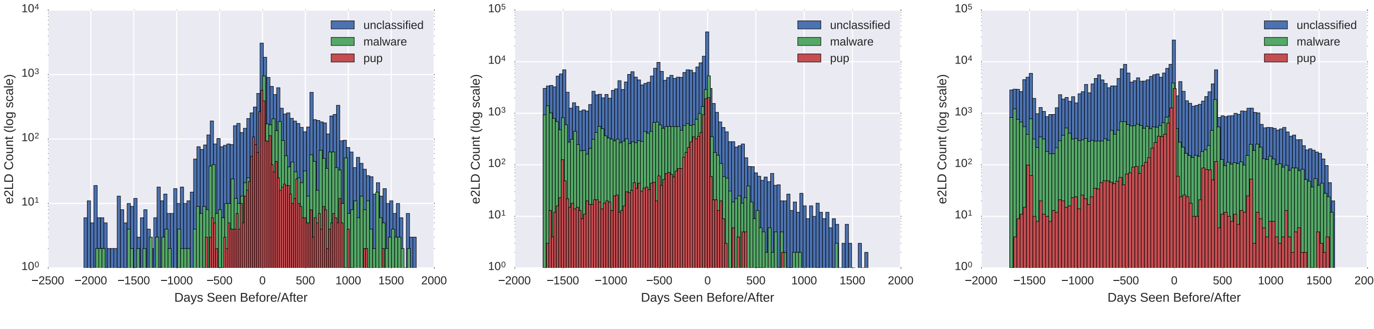


Fig. 5: Complete list of *all* known CDN domains queried by malware samples.



(a) PBL

(b) Passive DNS

(c) Expired

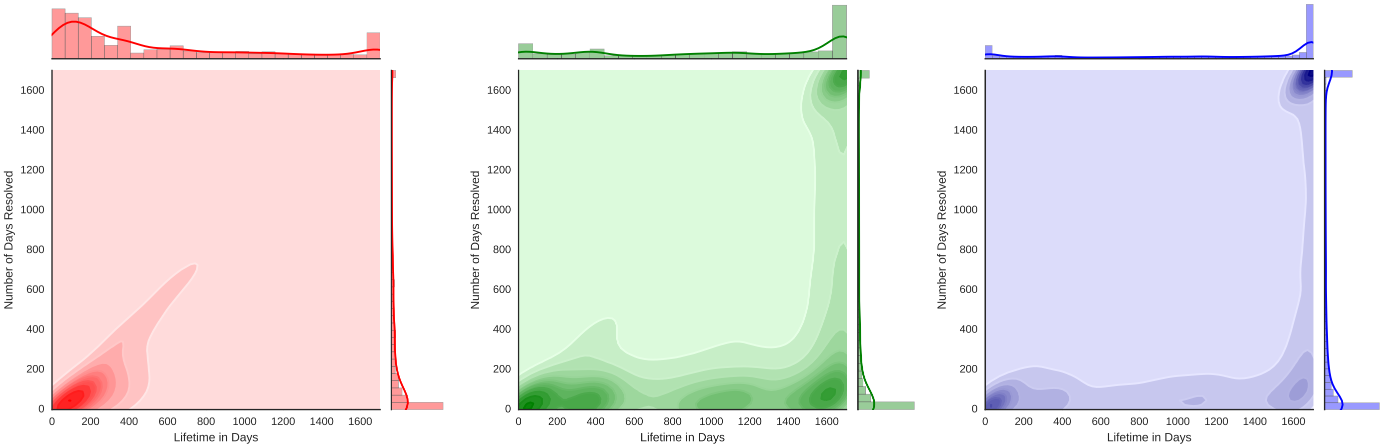
Fig. 6: Time difference between when a domain was first seen in passive DNS, public blacklists, or an expired domain list rather than through dynamic malware analysis.

insights into potential differences between these three classes of malicious software.

The figure shows that many domains were added to public blacklists only after we observed them in dynamic malware analysis traces. In particular, only 30% of the entries were added to blacklists before the domain was observed in our dynamic analysis dataset, while 20% of them were reported with a delay of over 500 days. *This result suggests that such delays could be largely reduced by relying on malware analysis to populate domains blacklists—possibly after applying a cleaning methodology like the one we described in Section III*. While it may seem reasonable to attribute this delay to the selection of blacklists used in this study, this result is consistent with previous work by Kührer et al [39] where domains were seen in passive DNS on average 384 days before appearing on a blacklist. Therefore, it is unlikely that the addition of other blacklists would profoundly affect this result. Additionally, reputation systems [18], [24] that rely on passive DNS have also demonstrated the ability to identify new threats more quickly than public blacklists. Furthermore, the observed delay between appearing in passive DNS and on public blacklists also lends credence to the idea of proactively detecting and blocking abuse at the time of domain registration as proposed by Hao et al. [32].

Next, we compare the date when we first observed a malware domain resolve in passive DNS with the date when the same domain was first observed in a dynamic malware analysis trace. By computing the difference between these two dates, we can determine how quickly new malware threats are discovered and analyzed by the security community. Figure 6b shows whether a malware domain was first seen in passive DNS or in a network trace derived by the dynamic analysis of a malware sample. Points less than zero on the x-axis indicate that a domain was first seen in passive DNS, and points greater than zero mean that the malware discovery occurred before the first observed network resolution in passive DNS.

The figure shows that the PUP-related domains are active an average of 192 days before we get to dynamically analyze the corresponding samples. This may be expected, as PUP relies on infrastructure that is more stable and long-running. However, we can see that popular malware families also follow a similar but less extreme pattern. This result is more surprising because for most of these domains the difference is very significant—with discovery delays reaching 623 days on average. Lastly, the domain names associated with the unclassified category follow the most interesting distribution.



(a) PUP

(b) Malware

(c) Unclassified

Fig. 7: Joint distribution of domain lifetime and resolution frequency observed in passive DNS for PUP, Malware, and Unclassified domains.

While many appear first in the passive DNS traces (left side of the Figure 6b), this category completely dominates the right tail of the graph—representing domains that were seen in passive DNS only months after we observed them in our sandboxes.

Overall, by combining the three classes together, we discovered that 302,953 malware domains were active at least two weeks—in some cases many months—before the corresponding malware samples were analyzed. *Therefore, while we previously showed that dynamic analysis systems could be used to improve current blacklists, our results also show that blacklists built from dynamic malware analysis will still be unaware of potential threats for several weeks or even months.*

The surprising nature of this result prompted us to perform additional analysis. Thus, we used our dataset of historic domain expirations to verify that a given domain was not used in the wild before expiring and being re-registered for malicious purposes. Figure 6c shows a similar pattern as the one in Figure 6a. The large peak around zero is likely a result of changes made to the domain infrastructure for unpopular or unused expiring domains. Such changes would likely result in DNS traffic to a parked or suspended page during the registrar's expiration grace period. Despite our extensive filtering efforts, the left tail in the graph remains significant. This can be partially explained by malware relying on benign infrastructure, such as dynamic DNS and CDN providers already mentioned in Section V-A. Another possible explanation is that the long tail is an artifact of a long setup phase for malware before it is released into the wild. During this phase, malware authors may age the domain and point it to benign infrastructure to build up positive reputation to help evade detection later. However, in the case of expired domains, this step is unnecessary because the domain will inherit the residual trust associated with the domain [43]—eliminating the need for a long aging phase. As shown in Figure 6c, we *still* see a long delay between last expiration of a domain and first discovery of an associated malware sample. Thus, one explanation could be that there is a long delay between a domain expiration and re-registration. In fact, this very behavior was seen in a study of spam related domain names by Hao et al. [33]. Their research showed that many registrations came hundreds of days after expiration, but abuse was observed less than three months after re-registration in most cases. Since we removed spam related malware samples in our extensive filtering step, this could suggest that the same behavior employed by spammers may also be used by other malware authors. However, it is also possible that domains are registered quickly after expiration and immediately used for abuse. Another potential explanation may be that our feeds are not the first to see new samples. However, since we are also considering first seen date from VirusTotal, it seems unlikely that the addition of other feeds that would dramatically reduce the first seen date for the malware samples in our dataset.

2) Domain Lifetime: Finally, we look at the lifetime of the malware domains using Figure 7, which shows a joint density distribution between the number of days a domain was resolved and the lifetime of that domain in days for PUPs, malware, and unclassified samples. We define the "lifetime" as the difference between the first and last seen dates for each of these domains in passive DNS.

The joint distribution makes it easy to infer not only how often a domain resolves but also for how long. In Figures 7b and 7c, we notice that there are three hotspots that correspond to the most prevalent resolution behaviors for domains in malware and unclassified malicious software. In the bottom left, we see that there are a large number of domains that are short lived and rarely resolved. A second hotspot, in the top right corner of both figures, corresponds to domains with the exact opposite behavior—long lived and frequently resolved. These domains were regularly queried for the entire duration of our experiments. Finally, in the bottom right of both figures, we observe a third pocket of domains that have a long lifetime but infrequent resolution. In this case, we observed only a few queries that were often years apart.

If we focus our attention on Figure 7a, which shows the lifetime patterns for PUP domains, we observe a completely different distribution. This is the consequence of two phenomena. First, we have seen the prevalence of PUP domains rise over the last two or three years, and this fact justifies the bounding of the joint distributions in the $[1,000, 1,000]$ region. The second reason has to do with the seemingly intense and continuing resolution of PUP related domains—which manifests as a higher density along the diagonal. We believe that this is a result of organizations failing to block PUP domains and end-point security engines that do not manage to remediate PUP infections. As Figure 7a shows, this gives PUPs a significantly different DNS resolution profile. On the other hand, the unclassified domains shown in Figure 7c follow a very similar pattern to the malware domains shown in Figure 7b. This likely indicates that most unclassified domains are very likely malware domains.

Summarizing, Figure 7 makes it clear that the all three types of domains frequently have long domain lifetimes, and many of those domains are frequently looked up. Since we showed that most domains were only resolved by a single sample in Section V-A1, this suggests that many samples remain active on the Internet for extended periods of time.

VI. Infrastructure Analysis

In this section, we analyze the hosting infrastructure for the domains resolved by the malware samples in our dataset. In particular, we want to investigate whether certain IP ranges appear more often the others, what are the reasons behind this choice, and how the global infrastructure picture evolved over time.

Figure 8 shows a histogram of the number of samples with domains (after filtering) resolving into a given /24 subnet, for each year between 2012 and 2015. In each plot, we observe spikes indicating that certain subnets are resolved by a very large number of samples during that year, from hundreds of thousands of samples in 2013 and 2014, up to peaks of few million samples in 2012 and 2015. Some of the spikes can be observed on multiple years e.g., `66.150.14.0/24` (Akamai) in 2012–2013 and `148.81.111.0/24` (NASK Polish CERT) in 2014–2015, while the majority appears in a single year.

We can assign spikes to specific families by analyzing the e2LDs that resolved to those ranges and use the mapping of e2LD to family produced by our classification in Section IV. This analysis reveals that there are three different reasons behind these spikes.

The largest group corresponds to spikes caused by specific malware families reusing the same subnet (sometimes even the exact same IP address) for long periods of time. Those families correspond to some of the Top-10 families by number of samples in Table III. The majority of these spikes are due to PUP families (`hotbar`, `domaiq`, `firseria`, `multiplug`) although we also observe some spikes due to polymorphic malware like `vobfus`. For example, three of the top four spikes in 2015 correspond to Amazon EC2 ranges and are all due to e2LDs belonging to the `multiplug` PUP family, which seems to have migrated its hosting infrastructure to EC2 in 2015. Similarly, the top spike in both 2012 and 2013 is caused by the `Hotbar` PUP family. This family used the Akamai CDN in 2012–2014 to host its infrastructure and therefore caused multiple spikes in different Akamai IP ranges. While we have observed a large number of benign domains resolving to the Akamai ranges, after our filtering in Section III it was simple to manually recognize the Akamai spikes in 2012–2014 and associate them to the `Hotbar` family.

The second group of spikes corresponds to sinkholes used to redirect resolutions of malicious domains after intervention. The most visible spike in this category is `148.81.111.0/24` in 2014–2015, which is due to `sinkhole.cert.pl`, used by NASK since 2014 to sink resolutions of the Polish domains used by the Virut botnet. Our dataset contains 123 Virut e2LDs resolving to this sinkhole being contacted by over 1M Virut samples in 2015. Another example of sinkhole-related spike is `0.0.0.0/24` in 2014, which is caused by a Microsoft-lead intervention on domains used by the `no-ip` dynamic DNS provider [3].

The third group of spikes is due to multiple, rather than a single, malware family. These indicate hotbeds of abuse and appear to keep changing over time. They may correspond to hosting providers that, over a certain time frame, had a more open policy on acceptable behavior. One such spike is on the right of the 2015 plot and corresponds to `Clara.Net` (`195.22.26.0/24`), a Portuguese hosting provider that hosted many domains associated to the `sality`, `wapomi`, `ramnit`, and `techsnab` malware families. Another example is the on the right side of the 2014 plot and corresponds to `ChinaNet` (`218.92.221.0/24`), where we observe domains, among others, of the `frethog` and `karnos` malware families.

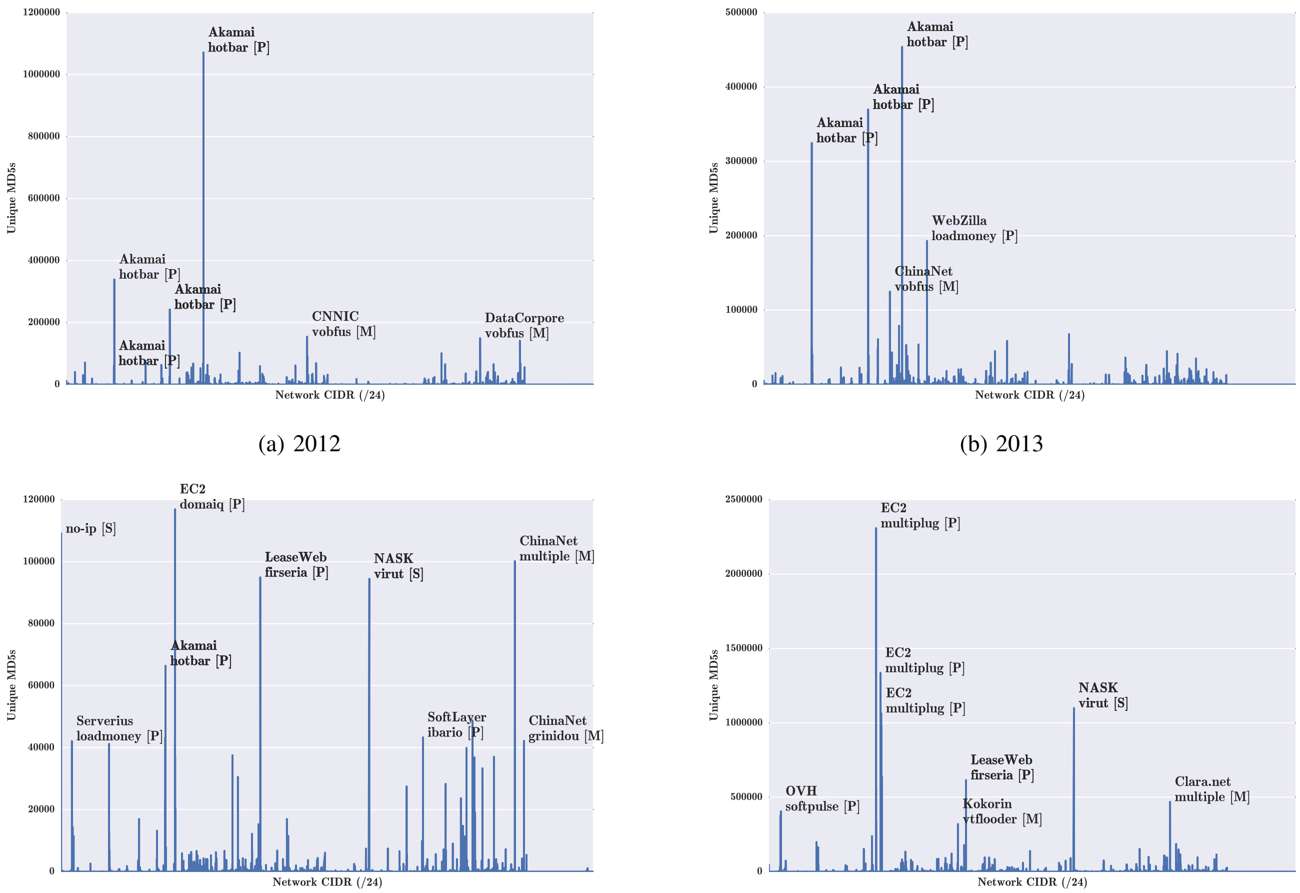
Regarding hosting providers, the top spikes in 2012 and 2013 correspond to Akamai (due to `hotbar`) and ChinaNet (`vobfus`). There is also one spike in 2013 due to `loadmoney` in the WebZilla cloud hosting service. As an interesting observation, starting from 2014 most spikes occur on IP ranges belonging to cloud hosting providers—most notably EC2, LeaseWeb, OVH, Serverius, and SoftLayer. One exception is a 2015 spike due to the `vtflooder` malware that resolved to `91.223.216.0/24`, which is registered to a private Ukranian person that uses a Gmail abuse email address.

Our analysis shows that the large spikes are dominated by PUP families and can last for multiple years indicating that PUP utilize seemingly stable IP infrastructure. This may indicate that popular cloud hosting providers like Amazon EC2, LeaseWeb, or OVH, and CDNs like Akamai, where PUP spikes happen, may not have the same policies towards banning PUP that they use for malware.

VII. DGA Malware

In the previous section, we provided an extensive analysis of the evolution of malware network infrastructure based on successful DNS resolutions. We now focus on a different aspect of malware behavior: the presence and impact on our dataset of domain name generation algorithms (DGAs).

Since the vast majority of DGA-related queries do not resolve to a valid IP address, to perform this analysis we first need to reintroduce in our dataset the failed (NXDomain) resolutions we filtered out in Section III. Since 2011, over 12.5 million malware samples in our dataset produced at least one



(a) 2012

(b) 2013

(c) 2014

(d) 2015

Fig. 8: Histograms of number of samples resolving domains that point to /24 subnets. Spikes are annotated with the owner of the IP range, the family that contacted it, and a letter indicating whether IPs are associated with malware(M), PUP(P), or a sinkhole (S).

NXDomain during their execution. The cumulative distribution function (CDF) in Figure 9 shows how half of the malware executions have less than two NXDomain resolutions and only 950,644 have over five.

To identify DGA-generated domains, we check if the domains in our malware executions dataset appear in the DGArchive [12], which comprises 50M domains generated by the DGAs of 66 malware families. Table V summarizes the overlapping of the DGArchive domains with both failed and successfully resolved domains in our malware executions dataset. For each family, it first shows how many e2LDs in the DGArchive are observed among the 6.8 M domains in our dataset before any filtering (i.e., including NXDomain queries). Then, it shows how many e2LDs from the DGArchive are observed among the 327 K domains remaining after all filtering.

According to the DGArchive, at least 44% (3 M) of all the e2LDs observed in our malware executions, regardless whether they successfully resolved or not, were generated by DGAs. This percentage is a lower bound since the DGArchive likely misses some DGA families, and also some variants that modify the DGA algorithm or its seed. After filtering, at least 17% of remaining e2LDs come from DGAs.

Our malware executions contain DGA e2LDs for 42 out of 66 (64%) families in the DGArchive. This number highlights the large coverage of our dataset. Of those 42 DGArchive families, 4 are variants of another family (e.g., pykspa and pykspa2), which we group into the 38 families in Table V. The large majority of DGA domains (82%) come from `virut`, followed by `pykspa` (6%) and `necurs` (4%). After filtering, successfully resolved `virut` DGA domains correspond to 12% of all unfiltered domains, followed by `zbot-gameover` (4%), and `ramnit` (0.5%). While `virut` is the most common DGA family in our dataset (and is still very active despite the 2014 takedown), if we normalize by number of samples in the family, we observe that `virut` resolves 1.8 e2LDs per sample, well below the most aggressive families: `emotet` (82 e2LDs/sample), `murofet` (68), and `cryptolocker` (53).

The table shows that only 1.8% of DGA domains successfully resolved. It also emphasizes different DGA behaviors by the malware families. For example, while we observe over 110K domain names queried by `necurs`, only one of them

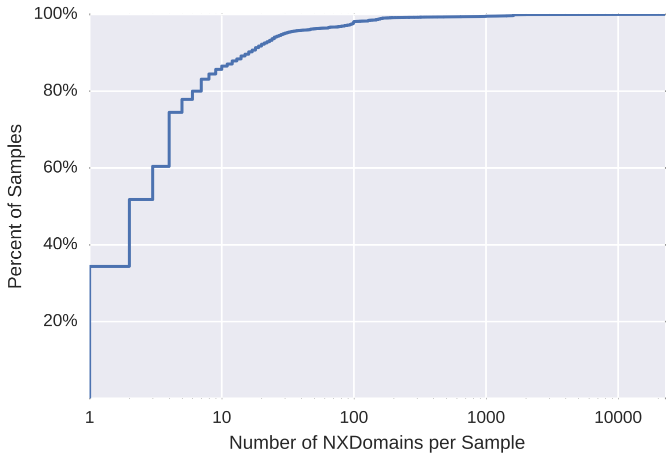


Fig. 9: Cumulative distribution function (CDF) for the number of NXDomains seen in malware samples in our datasets.

was active during our analysis. This indicates lack of pressure by defenders so the family can keep using the same domain. On the contrary, in other families like `gameover` roughly 30% of the queried domains resolved to a registered IP address.

The 24 families in the DGArchive that we do not observe are likely due to two reasons. First, some malware use the DGA as a backup option in the malware's communication strategy, i.e., only used if the primary C&C domains fail. This may not happen during the few minutes the samples are executed in the sandbox. Second, for some malware like `TDSS/TDL4`, the DGA is only used in specific components, i.e., monetization through ad abuse. Thus, we may not see DGA domains in our traces because the malware samples have not run long enough to be monetized.

Finally, in our dataset we also observe five candidate DGA families, not included in the DGArchive, which perform large numbers of NXDomain queries and have random-looking domains (`fosniw`, `shiz`, `softpulse`, `upatre`, `wapomi`).

VIII. Spam Related Malware

A common way to monetize malware is as distributed mechanism for sending spam. As discussed in Section III, we excluded spam related malware from our analysis because it is outside the scope of this work, and we did not wish to influence our analysis with network activity that is characteristic of typical spam behavior. Since there is already plenty of work that studies spam in great detail, we provide the following discussion only to provide more insight into the spam related samples we excluded from our analysis.

Prior to filtering, we identified a number of malware samples in our dataset that appeared to be involved in spam behavior. After gathering a set of domains used as MX records or that were likely to be associated with mail related activity, we built a collection of 405,742 malware samples that contacted these domains. This collection represents malware samples that were potentially involved in spamming activity. As seen in Figure 10, we noticed that these samples could be seen contacting hundreds to thousands of unique domains.

#	Family	Before Filtering	After Filtering
1	virut	2,477,628	40,452
2	pykspa	189,644	180
3	necurs	110,092	1
4	suppobox	72,476	4,677
5	tinba	52,463	682
6	gameover	24,325	7,083
7	emotet	23,500	96
8	pushdo	13,170	17
9	ranbyus	12,922	7
10	nymaim	12,490	148
11	simda	12,348	590
12	murofet	9,295	20
13	qakbot	4,130	119
14	ramnit	3,560	418
15	cryptolocker	2,912	89
16	conficker	1,710	465
17	sisron	1,394	1
18	oderoor	622	3
19	matsnu	525	130
20	dircrypt	510	53
21	tempedreve	204	20
22	banjori	200	1
23	feodo	192	13
24	urlzone	77	18
25	tsifiri	59	58
26	torpig	53	2
27	ramdo	49	27
28	gspy	49	0
29	bamital	48	2
30	bedep	44	5
31	hesperbot	37	2
32	fobber	31	2
33	gozi	24	8
34	bobax	23	0
35	proslikefan	12	1
36	darkshell	10	3
37	redyms	2	2
38	xxhex	1	1
All		3,026,831	55,396

TABLE V: DGA e2LD in the DGArchive [12] resolved in the malware executions in our dataset.

This observation can also be seen in Table VI that shows the ratio of MX lookups per malware sample. With the exception of Sality, the top 25 malware families queried at least 40 MX records. Since most malware we studied contacted less than 10 domains, this is an interesting characteristic of many of these samples.

Figure 10 shows the number MX lookups and mail related DNS queries for each potentially spam related malware sample. Notice that there is an interesting peak around 100 domains. After associating 5,239 samples with this spike, we were able to obtain VirusTotal reports for 909 (17.4%) of these samples; 892 (98.1%) were instances of Mydoom [77] malware—a compute worm first sighted in January 2004 and used by e-mail spammers to send mail from infected hosts. As noted in Section II, our earliest malware feeds start in January 2011, which is a full seven years after the first sighting of the Mydoom e-mail worm, and despite being known for over a decade, we still saw active Mydoom variants in our malware dataset as recently as August 2015. The long lifetime of this particular malware family is interesting because it suggests that even older malware is effective for spam related activity.

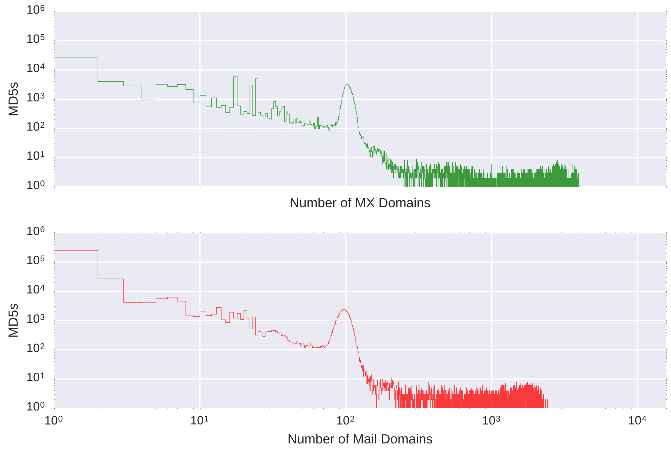


Fig. 10: Shows a histogram of the number of MD5s associated with each spam related domain in our filtering set.

Rank	Family	Samples	MX	Ratio
1	mydoom	82.0 K	6.0 M	72.8
2	hlux	1.7 K	3.5 M	2047.2
3	zbot	1.2 K	1.1 M	928.2
4	fareit	953	403.3 K	423.2
5	kelihos	446	388.0 K	870.1
6	winlock	171	373.3 K	2183.0
7	upatre	58	100.6 K	1734.0
8	zusy	54	92.0 K	1702.5
9	sality	23.6 K	86.2 K	3.7
10	tofsee	523	81.6 K	156.1
11	slym	38	63.7 K	1662.2
12	agentb	58	60.6 K	1045.2
13	tedroo	1.3 K	57.0 K	44.6
14	glupteba	21	45.1 K	2149.0
15	mikey	35	44.7 K	1276.2
16	bredolab	88	42.4 K	481.7
17	vblv	18	38.1 K	2115.8
18	yakes	85	35.9 K	422.4
19	tinba	13	34.8 K	2677.8
20	waledac	44	32.2 K	732.9
21	pwszbot	15	29.9 K	1992.4
22	ceeinject	19	26.0 K	1367.5
23	dorifel	195	24.8 K	127.5
24	zboter	10	23.5 K	2355.2
25	staser	2	21.6 K	10786.0
Top 25		112.7 K	12.8 M	113.3

TABLE VI: Top 25 spam families (filtered) ranked by number of MX lookups.

IX. Related Work

This work spans a number of different research areas each with a wealth of prior work. Therefore, this related work puts our research into context and also provide comparisons, where possible, with existing work.

A. *Malware Infrastructure*

A number of works have studied the malicious infrastructure used to distribute malware. For example, Rossow et al. [64] performed a large scale analysis of malware downloaders and their network infrastructure. Using the analysis traces of samples belonging to 23 downloader families, the authors discovered that 20% of the C&C servers remain operable long-term. In particular, the authors identified 2,942 C&C domains, which resolved to 861 IP addresses hosted in a variety of different ASs. Moreover, they found that malware infrastructures regularly migrate among different domains, while keeping a redundant presence in several different providers. In our experiments we are not able to distinguish C&C from other forms of malware traffic. However, in Section VI we discuss several characteristics of the hosting infrastructure of the domains resolved by the samples in our dataset and we found that some families use the same set of IP addresses for very long periods of time. In particular, PUP seem to have extremely stable infrastructures.

Wang et al. [75] built honeyclients to find drive-by download websites that exploit browser vulnerabilities. Similarly, Moshchuk et al. [51] used honeyclients to crawl over 18 million URLs, finding that 5.9% contained drive-by downloads. Provos et al. [61] studied the prevalence of drive-by downloads and the redirection chains leading to them, finding that 67% of the malware distribution servers were hosted in China [60]. These papers shed light on other aspects of the malware infrastructure, for instance by pointing out that malicious files are often hosted on multiple servers reachable by many different URLs at the same time. While malware samples can also occasionally contact infected web pages (e.g., to download additional components) our study focuses on the domains contacted by malware after a system has been infected and not on their drive-by download infrastructure. In a study closer to our work, Polychronakis et al. looked at the network behavior of malware distributed by drive-by downloads [59] using lightweight protocol responders. However, the authors focused on the purpose of the network activity (e.g., reporting home, data exfiltration, or joining a botnet) more than on the infrastructure and domains used by the malware authors. Active probing techniques have been proposed to measure the size of the server infrastructure for specific malware families [57], [79].

Another relevant line of work studies rogue networks and autonomous systems hosting unusually large amounts of malicious activity. To this end, Fire [69] used approximately one year of data from the Anubis system and Shue et al. [68] a dataset collected over a period of one month.

More recently, researchers investigated the use of cloud hosting providers as infrastructure for malware. These studies were conducted either by performing some form of active probing [56], [74] or by mining the information extracted from dynamic analysis sandboxes [30]. In this last case, the authors analyzed over 1M malware samples which connected to at least one publicly routable address on the Amazon EC2 Cloud. While this work did study network infrastructure, the results were limited to the Amazon cloud and required a considerable amount of manual analysis to separate and classify the different types of communication. By using our larger datasets we were able to confirm this trend and observe a similar effect on a more global scale, affecting multiple cloud providers - in particular from 2014. We also found that PUP families are the ones that rely the most on this type of stable infrastructure.

B. PUP Infrastructure

Recent work has studied the prevalence of PUP [38], [72], its distribution through pay-per-install (PPI) services [37], [73], and its detection [40], [41]. Thomas et al. [72] showed that ad-injectors affect 5% of unique daily IP addresses accessing Google. They also measured that Google's Safe Browsing generates three times as many detections for PUP as for malware [73]. Kotzias et al. measured that 54% of 3.9 M hosts they examined had PUP installed [37] and that PUP dominates so-called malware feeds [38]. Kwon et al. detect PUP and malware distribution using graph-based approaches leveraging machine learning [40] and temporal properties [41]. While some of these works explore PUP-related domains, none of them analyze properties of the PUP domain and server infrastructure. Most related are the categorization of the top 15,000 pages driving traffic to PPI downloaders by Thomas et al. [73] and the analysis by Kotzias et al. [37] of the top 20 e2LDs from where PUP is downloaded. Compared to our work, those analyses cover different time periods, only a fraction of PUP domains, and more importantly do not explore infrastructure properties such as PUP domain lifetime and hosting. In summary, we believe we are first to analyze the properties of the PUP domain and server infrastructure.

C. Longitudinal Malware Studies

While our study focuses exclusively on the network infrastructure, other researchers investigated the behavior extracted from dynamic analysis sandboxes to study other characteristics of malware samples.

One of the first attempts in this direction was performed by Bayer et al. [23] using almost 1M samples collected until 2009 by the Anubis platform. Interestingly, in this early study the authors reported that only 47.3% of the samples that showed some network activity also performed a DNS query—which succeeded in over 90% of the cases. Lindorfer et al. [44] performed a similar experiment focusing on the Android malware landscape. In this case, the authors reported that 99.91% of Android malware performed DNS queries, with roughly one third failing to resolve—suggesting an increasing adoption of domain generation algorithms (DGAs). While these studies provided interesting data points, they were performed at a distance of five years on datasets over 25 times smaller than the one used in this paper.

Several other papers have analyzed the use of DGAs, which can be used by botnets to bypass domain blacklists. Kolbitsch et al. [36] used binary code reuse [26] techniques to extract the DGA of the Conficker.A botnet and use it to compute the set of domains used on a given date. A related reverse-engineering approach was used by Plohmann et al. [58] to perform a large-scale analysis of DGAs used in malware. A different approach to detect and analyze DGAs is Pleiades [20], which monitors unsuccessful DNS resolution requests from recursive DNS servers in large networks. Our experiments confirm the wide use of DGA, and find that up to two thirds (67%) of the fully qualified domain names queried by malware failed to resolve to a valid address.

D. Internet Reputation

Network level analysis of malicious behavior offers a complementary means of characterizing and mitigating malware. For example, a popular method of preventing or limiting the spread of malware is the use of Internet blacklists. IP blacklists provide a list of known bad actors in the form of IP addresses which network operators can subsequently block; however, the use of DNS to build malicious network infrastructure has grown due to its resilience against IP blacklisting [65], [67].

Consequently, a significant amount of work has focused on analyzing network abuse at the DNS level [27], [31], [33], [35], [45], [76]. This has led to the creation of systems that are able to detect malicious domains through the use of passive DNS monitoring and machine learning [18], [19], [24]. Ultimately, these systems allow network operators to assemble DNS blacklists of malicious and suspicious domains in order to detect and prevent malicious activity on the network.

This has led researchers to study the usefulness of such blacklists. Metcalf et al [46] performed a comparison of 86 different internet blacklists—of both varying category and type—over a span of 30 months. Unfortunately, the inventory of blacklists appears to be partially anonymized, and some blacklists were collected for as little as three months. Therefore, it is difficult to compare these results directly with the blacklists used in this work. The major findings from their work showed that there is little overlap between lists for fully qualified domain names or IP addresses and, when there is overlap, no lists consistently outperforms another. Kührer et al. evaluated the effectiveness of 19 malware blacklists [39] collected over 2 years by classifying the entries (i.e.,non-existent, parked, or sinkholed), measuring the completeness, and testing the reaction time of each blacklist. When compared against a dataset of 300K samples, they show that the union of all blacklists contains 70% of domains detected per family. Furthermore, 58% of domains were seen in their passive DNS an average of 334 days before appearing on a blacklist. Our study shows similar results over a much longer observation period and with almost 100 times more malware.

Finally, Rajab et al. proposed a content-agnostic malware protection system based on binary and IP/DNS reputation to address the shortcomings of both blacklists and whitelists [63], and other works have been proposed that leverage network related reputation data for malware detection [34], [62].

X. Conclusion

After carefully filtering 26.8 million network traces obtained from dynamic malware execution, we are able to make several observations about the characteristics and temporal network properties of malware domains. First, we show that dynamic analysis traces should be carefully curated because they often contain a great deal of noise. To help with this challenge, we detail a rigorous methodology that analysts can use to remove potential noise from such traces. Next, potentially unwanted programs (PUPs) are not only on the rise, but surprisingly, they utilize seemingly stable IP infrastructure. In fact, we show that several hundred thousands PUP samples use the same network infrastructure over an entire year. Finally, our analysis shows that malware appears to add marginal detection benefits when trying to build early warning systems based on its network communication. We discovered that 302,953 malware domains were active at least two weeks—in some cases many months—before the corresponding malware samples were dynamically analyzed. This means that malware domain blacklists have limited detection value as malware tends to rapidly churn through domain names—yielding a very high rate of domain-level polymorphism.

XI. Acknowledgements

We are grateful to Daniel Plohmann for his help with the DGArchive and to VirusTotal for their support.

This material is based upon work supported in part by the US Department of Commerce grant 2106DEK, National Science Foundation (NSF) grant 2106DGX and Air Force Research Laboratory/Defense Advanced Research Projects Agency grant 2106DTX. This research was also partially supported by the Regional Government of Madrid through the N-GREENS Software-CM S2013/ICE-2731 project and by the Spanish Government through the DEDETIS grant TIN2015-7013-R. Any opinions, findings, conclusions, or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the US Department of Commerce, National Science Foundation, Air Force Research Laboratory, or Defense Advanced Research Projects Agency.

References

[1] DNSDynamic - Absolutely Free Dynamic DNS. https://www.dnsdynamic.org, 2016.

[2] Alexa. http://www.alexa.com/, 2007.

[3] Millions of dynamic DNS users suffer after Microsoft seizes No-IP domains. http://arstechnica.com/security/2014/06/millions-of-dymanic-dns-users-suffer-after-microsoft-seizes-no-ip-domains, 2014.

[4] Domain Blacklist: abuse.ch. http://www.abuse.ch/, 2015.

[5] Domain Blacklist: Blackhole DNS. http://www.malwaredomains.com/wordpress/?page_id=6, 2015.

[6] Domain Blacklist: driveby. http://www.blade-defender.org/eval-lab/, 2015.

[7] Domain Blacklist: hphosts. http://hosts-file.net/?s=Download, 2015.

[8] Domain Blacklist: itmate. http://vurl.mysteryfcm.co.uk/, 2015.

[9] Domain Blacklist: sagadc. http://dns-bh.sagadc.org/, 2015.

[10] Domain Blacklist: SANS. https://isc.sans.edu/suspicious_domains.html, 2015.

[11] Malware Domain List. http://www.malwaredomainlist.com/forums/index.php?topic=3270.0, 2015.

[12] DGArchive. https://dgarchive.caad.fkie.fraunhofer.de/, 2016.

[13] DNS-BH - Malware Domain Blocklist. http://www.malwaredomains.com/?cat=140, 2016.

[14] DynDNS. http://dyn.com/remote-access, 2016.

[15] Find Domain Names for Your DynDNS Pro Plan. http://dyn.com/remote-access/domain-names, 2016.

[16] FreeDNS. https://freedns.afraid.org, 2016.

[17] VirusTotal - Free Online Virus, Malware, and URL Scanner. http://www.virustotal.com, 2016.

[18] M. Antonakakis, R. Perdisci, D. Dagon, W. Lee, and N. Feamster. Building a Dynamic Reputation System for DNS. In *Proceedings of the 19th USENIX Security Symposium (SECURITY)*, 2010.

[19] M. Antonakakis, R. Perdisci, W. Lee, N. Vasiloglou, and D. Dagon. Detecting Malware Domains at the Upper DNS Hierarchy. In *Proceedings of the 20th USENIX Security Symposium (SECURITY)*, 2011.

[20] M. Antonakakis, R. Perdisci, Y. Nadji, N. Vasiloglou, S. Abu-Nimeh, W. Lee, and D. Dagon. From Throw-Away Traffic to Bots: Detecting the Rise of DGA-Based Malware. In *Proceedings of the 21st USENIX Security Symposium (SECURITY)*, 2012.

[21] APNIC. Reverse DNS Delegation. https://www.apnic.net/manage-ip/manage-resources/reverse-dns, 2009.

[22] M. Bailey, J. Oberheide, J. Andersen, Z. M. Mao, F. Jahanian, and J. Nazario. Automated Classification and Analysis of Internet Malware. In *Proceedings of the 10th International Symposium on Recent Advances in Intrusion Detection (RAID)*, 2007.

[23] U. Bayer, I. Habibi, D. Balzarotti, E. Kirda, and C. Kruegel. A View on Current Malware Behaviors. In *Proceedings of the 2nd USENIX Workshop on Large-Scale Exploits and Emergent Threats (LEET)*, 2009.

[24] L. Bilge, E. Kirda, C. Kruegel, and M. Balduzzi. EXPOSURE: Finding Malicious Domains Using Passive DNS Analysis. In *Proceedings of the 15th Network and Distributed System Security Symposium (NDSS)*, 2011.

[25] A. Broido, E. Nemeth, and kc claffy. Spectroscopy of DNS Update Traffic. https://www.caida.org/publications/papers/2003/dnsspectroscopy/dnsspectroscopy.pdf, 2005.

[26] J. Caballero, N. M. Johnson, S. McCamant, and D. Song. Binary Code Extraction and Interface Identification for Security Applications. In *Proceedings of the 14th Network and Distributed System Security Symposium (NDSS)*, 2010.

[27] D. Dagon, M. Antonakakis, P. Vixie, T. Jinmei, and W. Lee. Increased DNS Forgery Resistance Through 0x20-bit Encoding: SecURItY viA LeET QueRieS. In *Proceedings of the 15th ACM Conference on Computer and Communications Security (CCS)*, 2008.

[28] C. Grier, L. Ballard, J. Caballero, N. Chachra, C. J. Dietrich, K. Levchenko, P. Mavrommatis, D. McCoy, A. Nappa, A. Pitsillidis, N. Provos, M. Z. Rafique, M. A. Rajab, C. Rossow, K. Thomas, V. Paxson, S. Savage, and G. M. Voelker. Manufacturing Compromise: The Emergence of Exploit-as-a-Service. In *Proceedings of the 19th ACM Conference on Computer and Communications Security (CCS)*, 2012.

[29] L. Gundert. Dynamic Detection of Malicious DDNS. http://blogs.cisco.com/security/dynamic-detection-of-malicious-ddns, 2014.

[30] X. Han, N. Kheir, and D. Balzarotti. The Role of Cloud Services in Malicious Software: Trends and Insights. In *Proceedings of the 12th International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment (DIMVA)*, July 2015.

[31] S. Hao, N. Feamster, and R. Pandrangi. An Internet Wide View into DNS Lookup Patterns. Technical report, Verisign Labs, 2010.

[32] S. Hao, A. Kantchelian, B. Miller, V. Paxson, and N. Feamster. PREDATOR: Proactive Recognition and Elimination of Domain Abuse at Time-Of-Registration. In *Proceedings of the 23rd ACM Conference on Computer and Communications Security (CCS)*, 2016.

[33] S. Hao, M. Thomas, V. Paxson, N. Feamster, C. Kreibich, C. Grier, and S. Hollenbeck. Understanding the Domain Registration Behavior of Spammers. In *Proceedings of the 2013 Internet Measurement Conference (IMC)*, 2013.

[34] L. Invernizzi, S. Miskovic, R. Torres, C. Kruegel, S. Saha, G. Vigna, S.-J. Lee, and M. Mellia. Nazca: Detecting Malware Distribution in Large-Scale Networks. In *Proceedings of the 18th Network and Distributed Systems Symposium (NDSS)*, 2014.

[35] J. Jung, E. Sit, H. Balakrishnan, and R. Morris. DNS Performance and the Effectiveness of Caching. *IEEE/ACM Transactions on Networking*, 2002.

[36] C. Kolbitsch, T. Holz, C. Kruegel, and E. Kirda. Inspector Gadget: Automated Extraction of Proprietary Gadgets from Malware Binaries. In *Proceedings of the 31st IEEE Symposium on Security and Privacy (OAKLAND)*, 2010.

[37] P. Kotzias, L. Bilge, and J. Caballero. Measuring PUP Prevalence and PUP Distribution through Pay-Per-Install Services. In *Proceedings of the 25th USENIX Security Symposium (SECURITY)*, 2016.

[38] P. Kotzias, S. Matic, R. Rivera, and J. Caballero. Certified PUP: Abuse in Authenticode Code Signing. In *Proceedings of the 22nd ACM Conference on Computer and Communication Security (CCS)*, 2015.

[39] M. Kührer, C. Rossow, and T. Holz. Paint It Black: Evaluating the Effectiveness of Malware Blacklists. In *Proceedings of the 18th International Syposium on Research in Attacks, Intrusions, and Defenses (RAID)*, 2014.

[40] B. J. Kwon, J. Mondal, J. Jang, L. Bilge, and T. Dumitras. The Dropper Effect: Insights into Malware Distribution with Downloader

Graph Analytics. In *Proceedings of 22nd ACM Conference on Computer and Communications Security (CCS)*, 2015.

[41] B. J. Kwon, V. Srinivas, A. Deshpande, and T. Dumitras. Catching Worms, Trojan Horses and PUPs: Unsupervised Detection of Silent Delivery Campaigns. In *Proceedings of the 20th Network and Distributed Systems Security Symposium (NDSS)*, 2017.

[42] I. S. S. Lab. Anubis - Malware Analysis for Unknown Binaries. http://anubis.iseclab.org, 2016.

[43] C. Lever, R. Walls, Y. Nadji, D. Dagon, P. McDaniel, and M. Antonakakis. Domain-Z: 28 Registrations Later. In *Procededings of the 37th IEEE Symposium on Security and Privacy (OAKLAND)*, 2016.

[44] M. Lindorfer, M. Neugschwandtner, L. Weichselbaum, Y. Fratantonio, V. Van der Veen, and C. Platzer. Andrubis - 1,000,000 Apps Later: A View on Current Android Malware Behaviors. In *Proceedings of the 3rd International Workshop on Building Analysis Datasets and Gathering Experience Returns for Security (BADGERS)*, 2014.

[45] C. Liu and P. Albitz. *DNS and BIND*. O'Reilly Media, 5th edition edition, 2006.

[46] L. Metcalf and J. M. Spring. Blacklist Ecosystem Analysis: Spanning Jan 2012 to Jun 2014. In *Proceedings of the 2nd ACM Workshop on Information Sharing and Collaborative Security (WISCS)*, 2015.

[47] Microsoft. Viewing Vobfus Infections from Above. https://trac.torproject.org/projects/tor/ticket/7349, 2013.

[48] P. Mockapetris. Domain names - concepts and facilities. RFC 1034 (Standard), Nov. 1987.

[49] P. Mockapetris. Domain names - implementation and specification. RFC 1035 (Standard), Nov. 1987.

[50] A. Mohaisen and O. Alrawi. AV-Meter: An Evaluation of Antivirus Scans and Labels. In *Proceedings of the 11th Detection of Intrusions and Malware, and Vulnerability Assessment (DIMVA)*, 2014.

[51] A. Moschuk, T. Bragin, S. D. Gribble, and H. Levy. A Crawler-based Study of Spyware in the Web. In *Proceedings of the 10th Network and Distributed System Security Symposium (NDSS)*, 2006.

[52] A. Moser, C. Kruegel, and E. Kirda. Exploring multiple execution paths for malware analysis. In *Proceedings of the 28th IEEE Symposium on Security and Privacy (OAKLAND)*, 2007.

[53] B. Moskowitz, D. Karrenberg, G. J. de Groot, and E. Lear. Address Allocation for Private Internets. https://tools.ietf.org/rfc/rfc1918.txt, 1996.

[54] Y. Nadji, M. Antonakakis, R. Perdisci, D. Dagon, and W. Lee. Beheading Hydras: Performing Effective Botnet Takedowns. In *Proceedings of the 20th ACM Conference on Computer and Communications Security (CCS)*, 2013.

[55] Y. Nadji, M. Antonakakis, R. Perdisci, and W. Lee. Understanding the Prevalence and Use of Alternative Plans in Malware with Network Games. In *Proceedings of the 27th Annual Computer Security Applications Conference (ACSAC)*, 2011.

[56] A. Nappa, M. Z. Rafique, and J. Caballero. Driving in the Cloud: An Analysis of Drive-By Download Operations and Abuse Reporting. In *Proceedings of the 10th International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment (DIMVA)*, 2013.

[57] A. Nappa, Z. Xu, J. Caballero, and G. Gu. CyberProbe: Towards Internet-Scale Active Detection of Malicious Servers. In *Proceedings of the 18th Network and Distributed Systems Symposium (NDSS)*, 2014.

[58] D. Plohmann, K. Yakdan, M. Klatt, J. Bader, and E. Gerhards-Padilla. A Comprehensive Measurement Study of Domain Generating Malware. In *Proceedings of the 25th USENIX Security Symposium (SECURITY)*, 2016.

[59] M. Polychronakis, P. Mavrommatis, and N. Provos. Ghost Turns Zombie: Exploring the Life Cycle of Web-Based Malware. In *Proceedings of the 1st USENIX Workshop on Large-Scale Exploits and Emergent Threats (LEET)*, 2008.

[60] N. Provos, P. Mavrommatis, M. A. Rajab, and F. Monrose. All Your iFRAMEs Point to Us. In *Proceedings of the 17th USENIX Security Symposium (SECURITY)*, 2008.

[61] N. Provos, D. McNamee, P. Mavrommatis, K. Wang, and N. Modadugu. The Ghost in the Browser: Analysis of Web-Based Malware. In *Proceedings of the 2nd USENIX Workshop on Hot Topics on Understanding Botnets (HotSec)*, 2007.

[62] B. Rahbarinia, M. Balduzzi, and R. Perdisci. Real-Time Detection of Malware Downloads via Large-Scale URL→File→Machine Graph Mining. In *Proceedings of the 11th ACM on Asia Conference on Computer and Communications Security (ASIACCS)*, 2016.

[63] M. A. Rajab, L. Ballard, N. Lutz, P. Mavrommatis, and N. Provos. CAMP: Content-Agnostic Malware Protection. In *Proceedings of the 17th Network and Distributed System Security Symposium (NDSS)*, 2013.

[64] C. Rossow, C. Dietrich, and H. Bos. Large-scale Analysis of Malware Downloaders. In *Proceedings of the 9th Conference on Detection of Intrusions and Malware, and Vulernability Assessment (DIMVA)*, 2012.

[65] P. Royal. Analysis of the Kraken Botnet. Technical report, Damballa Labs, 2008.

[66] M. Sebastián, R. Rivera, P. Kotzias, and J. Caballero. AVClass: A Tool for Massive Malware Labeling. In *Proceedings of the 19th International Symposium on Research in Attacks, Intrusions and Defenses (RAID)*, 2016.

[67] S. Shevchenko. Srizbi's Domain Calculator. http://blog.threatexpert.com/2008/11/srizbis-domain-calculator.html, 2008.

[68] C. Shue, A. J. Kalafut, and M. Gupta. Abnormally Malicious Autonomous Systems and Their Internet Connectivity. *IEEE/ACM Transactions of Networking*, 2012.

[69] B. Stone-Gross, Christopher, Kruegel, K. Almeroth, A. Moser, and E. Kirda. FIRE: FInding Rogue Networks. In *Proceedings of the 25th Annual Compuer Security Applications Conference (ACSAC)*, 2009.

[70] Symantec. W32.Virut. https://www.symantec.com/security_response/writeup.jsp?docid=2007-041117-2623-99, 2012.

[71] T. Taylor, X. Hu, T. Wang, J. Jang, M. P. Stoecklin, F. Monrose, and R. Sailer. Detecting Malicious Exploit Kits Using Tree-based Similarity Searches. In *Proceedings of the 6th ACM Conference on Data and Application Security and Privacy (CODASPY)*, 2016.

[72] K. Thomas, E. Bursztein, C. Grier, G. Ho, N. Jagpal, A. Kapravelos, D. McCoy, A. Nappa, V. Paxson, P. Pearce, N. Provos, and M. A. Rajab. Ad Injection at Scale: Assessing Deceptive Advertisement Modifications. In *Proceedings of the 36th IEEE Symposium on Security and Privacy (OAKLAND)*, 2015.

[73] K. Thomass, J. A. E. Crespo, R. Rastil, J.-M. Picodi, L. Ballard, M. A. Rajab, N. Provos, E. Bursztein, and D. Mccoy. Investigating Commercial Pay-Per-Install and the Distribution of Unwanted Software. In *Proceedings of the 25th USENIX Security Symposium (SECURITY)*, 2016.

[74] L. Wang, A. Nappa, J. Caballero, T. Ristenpart, and A. Akella. WhoWas: A Platform for Measuring Web Deployments on IaaS Clouds. In *Proceedings of the 2014 ACM Internet Measurement Conference (IMC)*, 2014.

[75] Y.-M. Wang, D. Beck, X. Jiang, R. Roussev, C. Verbowski, S. Chen, and S. King. Automated Web Patrol with Strider HoneyMonkeys: Finding Web Sites That Exploit Browser Vulnerabilities. In *Proceedings of the 10th Network and Distributed System Security Symposium (NDSS)*, 2006.

[76] D. Wessels, M. Fomenkov, N. Brownlee, and K. Claffy. Measurements and Laboratory Simulations of the Upper DNS Hierarchy. In *Proceedings of the 5th International Passive and Active Measurement Workshop (PAM)*, 2004.

[77] Wikipedia. Mydoom. https://en.wikipedia.org/wiki/Mydoom, 2016.

[78] C. Willems, T. Holz, and F. Freiling. Toward Automated Dynamic Malware Analysis Using CWSandbox. *IEEE Security & Privacy*, 2007.

[79] Z. Xu, A. Nappa, R. Baykov, G. Yang, J. Caballero, and G. Gu. AutoProbe: Towards Automatic Active Malicious Server Probing Using Dynamic Binary Analysis. In *Proceedings of the 21st ACM Conference on Computer and Communication Security (CCS)*, 2014.

[80] T.-F. Yen and M. K. Reiter. Traffic Aggregation for Malware Detection. In *Proceedings of the 5th International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment (DIMVA)*, 2008.

[81] J. Zhang, R. Perdisci, W. Lee, U. Sarfraz, and X. Luo. Detecting Stealthy P2P Botnets Using Statistical Traffic Fingerprints. In *Proceedings of the 41st International Conference on Dependable Systems and Networks (DSN)*, 2011.