Measuring Network Reputation in the Ad-Bidding Process

Yizheng Chen[1], Yacin Nadji[2], Rosa Romero-Gómez[2], Manos Antonakakis[2], and David Dagon[1]

[1] Georgia Institute of Technology, School of Computer Science,
yzchen@gatech.edu, dagon@sudo.sh
[2] Georgia Institute of Technology, School of Electrical and Computer Engineering,
{yacin,rgomez30,manos}@gatech.edu

Abstract. Online advertising is a multi-billion dollar market, and therefore a target for abuse by Internet criminals. Prior work has shown millions of dollars of advertisers' capital are lost due to ad abuse and focused on defense from the perspective of the end-host or the local network egress point. We investigate the potential of using public threat data to measure and detect adware and malicious affiliate traffic from the perspective of demand side platforms, which facilitate ad bidding between ad exchanges and advertisers. Our results show that malicious ad campaigns have statistically significant differences in traffic and lookup patterns from benign ones, however, public blacklists can only label a small percentage of ad publishers (0.27%), which suggests new lists dedicated to ad abuse should be created. Furthermore, we show malicious infrastructure on ad exchanges can be tracked with simple graph analysis and maliciousness heuristics.

1 Introduction

On-line advertisement is a complex ecosystem that enables one of the most prosperous Internet businesses. Naturally, it has become the target of abuse. Only in the last few years are we beginning to grasp the scale of the economic loss for advertisers from ad abuse [10, 15, 24, 28]. Using armies of compromised machines (i.e., botnets), sophisticated affiliate programs, and ad injection techniques, millions of dollars are stolen from advertisers. If we want to reduce abuse on the Internet, we will have to eliminate the monetization opportunities attackers use.

For almost a decade, security researchers and network operators have studied how to detect and stop advertisement abuse. The focus of past research efforts has been on detecting ad abuse at the edge (i.e., the infected host), at the egress point of a network, or "outside" of the ad ecosystem. Little is known, however, about the network policies that are being enforced *within* the ad ecosystem, especially during the ad bidding process. Advertisers do not want to display ads on low quality publishers that may include automated visits from adware and affiliate marketing entities, and thus they need to selectively respond to ad

bidding requests based on the reputation of the publishers. Unfortunately, little work has been done to measure reputation of publisher domains.

In this paper, we examine if open source intelligence data from the security community can be used to ascertain publisher reputation. To this end, we analyze anonymized ad bidding requests between a large demand side platform (DSP) in North America and six ad exchanges over a period of three months. Using open source intelligence from public blacklists and malware execution traces, we investigate the reputation properties of publishers in the *advertisement bidding process* (Section 5). Our study makes the following key observations:

- We explain the ad bidding process and measure it in detail to improve the network and security communities' understanding of the advertising ecosystem. These measurements include bidding request traffic from six large ad exchanges for request volume, publisher domains, and client distribution. We find that malicious publisher domains tend to be present on more ad exchanges and reach more clients than non-blacklisted publisher domains on average. These differences are statistically significant and suggest that reputation systems for advertisement publishers are possible.
- We identify that of all publisher domains seen in the DSP, 13,324 (0.27%) are on blacklists, which generate only 1.8% of bid requests, and 134,262 (2.74%) are queried by malware. This underestimates the amount of ad abuse based on other studies [14, 16], which has been measured as high as 30%. This also indicates that traditional sources of maliciousness used in the security community are insufficient to understand ad abuse seen from DSPs.
- Using graph analysis, we demonstrate how to track advertising infrastructure over time. To focus on potentially malicious campaigns, we use a simple suspiciousness heuristic based on open-source intelligence feeds. Using this technique, we identify case studies that show ad network domains support Potentially Unwanted Programs (PUP), rely on domain name generation algorithms, and are occasionally used to distribute malware.

2 Background

In this section, we briefly describe the key components of the ad ecosystem and the real-time bidding process.

2.1 Ad Ecosystem

Figure 1 gives an overview of the online advertising ecosystem. When a user visits a publisher webpage (step 1, Figure 1) its elements are loaded (step 2), during which the iFrame representing the *ad inventory* requests the ad server for an ad to display (step 3). The ad server asks for an ad from the ad network (step 4), and reports ad metrics for payment logging. An ad network can also sell ad inventories to an ad exchange (step 5). If an ad request cannot be fulfilled, it will be relayed to a Demand Side Platform provider (DSP) (step 6), and then advertisers who work with the DSP can purchase the impression (scenario A).

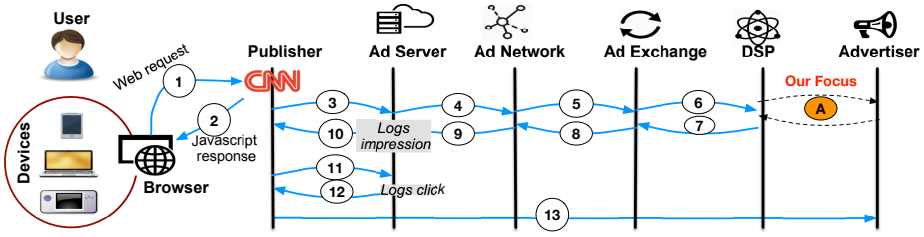


Fig. 1: An overview of the online advertising ecosystem.

The advantage of using a DSP is that advertisers will have access to multiple ad exchanges. In this paper, we focus on the vantage point of a DSP (scenario A).

The DSP, ad exchanges, and ad networks consolidate advertisers' audience target and budget information, and show the optimal ad back to the publisher's page (step 7 to 10). An impression is therefore fulfilled and logged. Impressions are often charged according to the CPM (Cost Per Mille, or cost per thousand impression). If the ad is clicked, the ad server will log it (step 11), and redirect the user (step 12) to the page of the advertiser (step 13). In such an event, the advertiser is charged for the click. The CPC (Cost Per Click) varies according to the keywords of the webpage and the user category.

Publishers can resell (*syndicate*) the ads to other publishers. In turn, these publishers can sell (*subsyndicate*) the ads further to other publishers. Syndication enables the ads to reach a wider audience. Thus, there can be several redirections among publishers before an ad request reaches the ad server (step 3).

2.2 Real-Time Bidding

Figure 2 shows a simplified view of the Real-Time Bidding (RTB) process. The JavaScript from the publisher page requests an ad through a *bid request*. In a request, the publisher includes information such as category of the page, size of the ad space, country, user's browser and OS version, cookie, etc., and sends it to the ad exchange (step 1).

Once the ad exchange receives the bid request from a seller, it then consolidates the request into seller site information (e.g., URL of the publisher page), device information, and user data. The ad exchange sends the bid request to its buyer applications (step 2), for instance, through a DSP.

After receiving the bid request, the buyer replies with a bid response containing the ad URL and the markup price (step 3). The RTB protocol typically waits for a fixed amount of time (e.g., 100ms) to collect bids, and then chooses the winning bid under the auction's rules (e.g., OpenRTB [27]). The ad exchange then notifies the winner and returns the ad to the publisher (step 4).

In the aforementioned example, the bid request comes from the publisher directly. Therefore, the publisher page is the *referrer* for the bid request. Very often, the bid request comes from the market place, where the original request was purchased and resold by many intermediaries. In that case, the *referrer* is

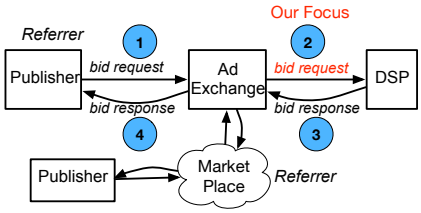


Fig. 2: A simplified view of the Real-Time Bidding process.

	Date Range	Size
DSP Traffic	12/10/14 - 3/24/15	2.61T
Blacklists	12/9/09 - 1/15/16	22G
Malware	1/1/11 - 11/17/15	136G
DNS	12/10/14 - 3/24/15	1.54T

Table 1: Summary of all datasets.

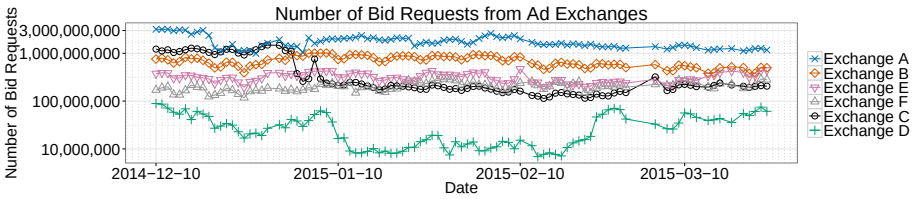


Fig. 3: Number of daily bid requests from ad exchanges seen in the DSP.

the last entity that sold the ad inventory to the ad exchange. Ad exchanges do not have visibility of the user-side publisher if the request comes from the market place. This is one of the challenges for ad exchanges to detect and stop fraud.

3 Datasets

In this section, we describe the datasets we obtained including Demand Side Platform provider (DSP) traffic, public blacklist data, and malware domain data. Table 1 provides a brief summary of the datasets.

3.1 DSP Traffic

The DSP provides ad bidding logs extracted from step 3 of Figure 2. The traffic is aggregated into eight fields per hour every day: the **ad exchange** that issued the bid request, the **publisher domain name** of the referrer URL, the **hashed IP address** of the user, the **country code** and **autonomous system number** of the IP address, the hourly **timestamp** of when the bid request was sent, and lastly the **number of bid requests** seen within the specific hour that match all the previous fields. Within the fields, the **publisher domain name** represents either the webpage that users saw, or the last traffic reseller before the bid request reached the ad exchange. Next, we describe DSP traffic using the volume of bid requests and publisher domain names.

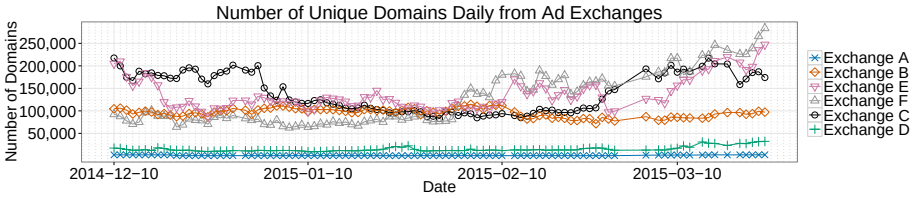


Fig. 4: Number of daily publisher domains from ad exchanges seen in the DSP.

Bid Request Volume It is reasonable to assume that for each bid request, some advertiser wins the bid eventually. Therefore, the bid request volume can be considered to be the number of ad inventories purchased and shuffled through the ad exchanges from the visibility of the DSP.

Figure 3 shows the bid request volume from six different ad exchanges from 12/10/2014 to 3/24/2015. One of these ad exchanges is ranked top five in market share. On average, there are 3.45 billion bid requests daily in total. Individually, Exchange A processed the most bid requests of all, with an average of 1.77 billion requests per day. Exchange B comes next, with an average of 695 million requests per day. In addition, Exchange E, Exchange F, and Exchange C received bid requests on the order of hundreds of millions. Finally, Exchange D had an average of 30 million bid requests daily, which fluctuated the most compared to other ad exchanges.

Comparing the volume of the last day from the DSP traffic (3/24/2015) with that of the first day (12/10/2014), there is a decline in the overall bid request volume from Exchange A (63.2%), Exchange B (34.3%), Exchange C (83.2%), and Exchange D (31.2%). However, the volume increased for Exchange E (18.34%) and Exchange F (64.26%). Our DSP confirmed that this was not a traffic collection problem but could not identify the root cause of these changes.

Publisher Domains The publisher domain field in the DSP traffic indicates the source of an ad request. It is either the publisher website where the ad will be shown, or the reseller domain redirected from some previous publisher.

An average of 391,430 total publisher domains were seen from all ad exchanges every day. Figure 4 shows the number of unique publisher domains from each ad exchange. Although Exchange A had the highest number of bid requests (Figure 3), it represented the lowest number of unique domains (average: 955) per day. It is likely that many of them are traffic resellers. For instance, `coxdigitalsolutions.com` is a subsidiary of Cox specializing in buying and selling digital media. It is the most popular publisher domain in Exchange A, generating more than 20% of all bid requests. The small set of publisher domains of Exchange A is quite stable. There were no new publishers in 39 days out of three months, and an average of 91 new publisher domains on the other days. Exchange D has the fewest bid requests and also had very few publisher domains, an average of 14,732 every day. If an ad exchange works with few publishers, it

is easier to provision them and block malicious traffic. On the other hand, it is harder to know the source of ad inventories from reseller publishers, meaning detection may need to happen at the reseller's perspective.

Two ad exchanges saw the largest number of new publisher domains. Exchange E had an average of 22,647 new publisher domains, while Exchange F had an average of 23,405 new publisher domains daily. Towards the end of March 2015 in Figure 4, there were as many as 35,794 new domains from Exchange E and 56,151 new domains from Exchange F. Both ad exchanges also increased the volume of bid requests during the same time period in Figure 3. The churn rates of the publisher domain names in these two ad exchanges were quite high. This presents a challenge for ad exchanges to track the reputation of new publishers.

Lastly, Exchange B had a stable number of publisher domains every day, on the order of 100,000. There was a decrease in the number of daily publisher domains seen from Exchange C around the end of 2014, and then the number increased again, reaching the 150,000 mark towards the end of March 2015.

3.2 Other Datasets

In order to measure reputation in the DSP bid request traffic, we also obtained other datasets that provide threat information, which includes public blacklists and dynamic malware execution traffic. Both provide insight into known abuse in the ad exchanges. We crawled seven public blacklists [2–5, 7, 8, 39] daily from 12/9/2009 to 1/15/2016. In total, 1.92 million unique domains appeared on the public blacklists. Dynamic malware execution feeds are from one university [20] and two industry partners. The binaries were each executed for five minutes in a controlled environment. We extracted date, malware md5, and the domain names queried during the execution of the binaries. The feeds are collected from 1/1/2011 to 11/17/2015. There are 77.29 million unique malware md5s, querying a total of 14.3 million domain names. We use *PBL* to denote the public blacklists dataset and *Md5* to denote the malware domains dataset.

Lastly, we collected DNS resolution data every day from a passive DNS repository in North America between 12/10/2014 to 3/24/2015. The dataset contains domain name, query type, and resolved data every day for A, NS, CNAME, and AAAA query types. We observed a daily average of 891 million unique mappings between domain names. On average, the DNS resolution dataset matches 71.56% of all publisher domain names seen in the DSP in the same day. Among the 28.55% publisher domains from DSP not seen in passive DNS, the majority of them are long tail content sites. For example, unpopular blog sites, user's own fantasy sport pages, customized lists pages, etc. Long tail content can be specific to certain users' interests and not commonly accessed across different networks. In full disclosure, this is perhaps the only not fully open source intelligence source we used in our experiments. However, commercial passive DNS offerings are very simple to obtain today [6]. We will use the resolution information to construct infrastructure graphs and track them over time in Section 6.

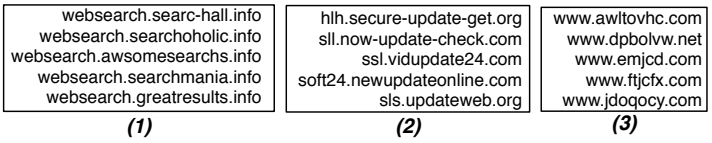


Fig. 5: Examples of blacklisted publisher domains seen in the DSP traffic.

4 Fraudulent Publisher Domains

In this section we provide examples of blacklisted publisher domains that generated ad bidding requests through the ad exchanges. These domains are from adware and affiliate marketing programs.

4.1 Case 1: PUP

Blacklisted publisher domains can be generated by Potentially Unwanted Programs (PUP) such as browser hijacker and pop-up ads.

Figure 5 (1) shows domain names of pattern `websearch.*.info` that are used by browser hijackers [22]. The adware forces the user to use a different search engine to steal impressions that would have otherwise been delivered through typical search engines (e.g., Google, Bing, Yahoo, etc.). The adware hijacks user search queries and makes ad bidding requests from these publisher domains to generate revenue.

Figure 5 (2) shows "update" domains used by pop-up ads. The adware shows pop-up ads that masquerade as fake updaters for legitimate software, such as Windows, Flash, and video players [23]. These publisher domains make ad bidding requests from pop-up windows generated by the adware.

4.2 Case 2: Affiliate Marketing

Blacklisted publisher domains may represent affiliate marketing domains. These affiliate domains request ads through ad exchanges on behalf of adware or malware. We manually analyzed network traces from dynamic execution of malware md5s that contained domains in Figure 5 (3). The malware uses fake referrers to send HTTP GET requests through domains in Figure 5 (3). Then the requests go through a chain of redirections until finally receiving an ad to generate revenue.

5 Measurement

We first discuss client IP location distribution in DSP traffic in Section 5.2. Then, we perform reputation analysis of publisher domains by correlating them with blacklists and malware domains in Section 5.3.

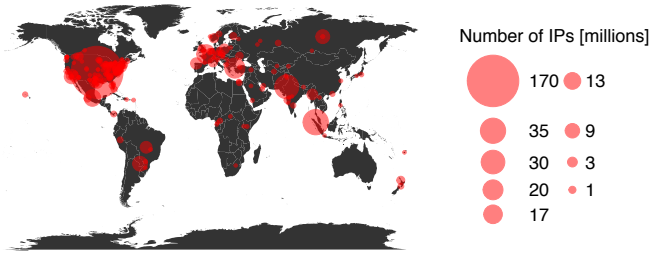


Fig. 6: Distributions of client IP address locations.

5.1 Summary of Findings

In summary, we found that:

- There are 13,324 (0.27%) known malicious domains generating bid request traffic through the ad exchanges in our datasets. On average, they generate 1.8% of overall bid requests daily, much less than previously published values [14, 16]. However, 68.28% of blacklisted domains were identified by public blacklists before they appeared in DSP traffic. This suggests traditional sources of maliciousness are valuable, but insufficient to understand ad-abuse from the perspective of DSPs.
- On average, blacklisted publisher domains tend to use more ad exchanges (average: 1.85) and reach more clients (average: 5109.47) compared to non-blacklisted domains (average ad exchanges: 1.43, average hashed client IP addresses: 568.78) (Section 5.3). This suggests reputation systems for ad publishers are possible.
- Contrary to the observation of blacklisted publisher domains, malware domains use a similar number of ad exchanges (average: 1.44), but are seen from more hashed client IP addresses (average: 2310.75), compared to publisher domains never queried by malware (average ad exchanges: 1.43, average hashed client IP addresses: 485.36). (Section 5.3)

5.2 Client Analysis

We observed 436 million hashed client IPs that sent bid requests for ads. According to information provided by the DSP, the hashed client IP addresses are from 37,865 different Autonomous Systems in 234 different countries.

Table 2a shows the top six countries where hashed client IP addresses reside. Nearly 40% of clients are located in the United States. Next, it is the United Kingdom with 8% of hashed IP addresses. The top six countries also include Germany (7.11%), Canada (4.82%), France (3.90%), and Mexico (2.98%). There is a long tail of 228 other countries for the remaining clients. Overall the top six countries account for 66.75% of all the hashed client IP addresses seen in DSP. Figure 6 shows the country distribution of hashed client IP address locations.

Table 2b presents the top six Autonomous System Names (ASNs) for hashed client IP addresses. The ASN distribution is less biased compared to the country

Country	Hashed IPs millions
US	174 (39.91%)
GB	35 (8.03%)
DE	31 (7.11%)
CA	21 (4.82%)
FR	17 (3.90%)
MX	13 (2.98%)
Other	103 (23.62%)
Unknown	42 (9.63%)
Total	436 (100.00%)

(a) Client Location

AS Names	Hashed IPs millions
Comcast	18 (4.13%)
AT&T	17 (3.90%)
Deutsche Telekom	14 (3.21%)
MCI	12 (2.75%)
Verizon	9 (2.06%)
Uninet	7 (1.61%)
Other	359 (82.34%)
Unknown	42 (9.63%)
Total	436 (100.00%)

(b) AS Name

Table 2: 2a: The top six countries for 66.75% of hashed client IP addresses. 2b: The top six Autonomous System Names for 17.66% of hashed client IP addresses.

distribution. Comcast, AT&T, and Deutsche Telekom are the top three ASNs, each with under 5% of all hashed IP addresses. There are 37,859 different ASNs in the long tail of the distribution, which contains 82.34% of all hashed IPs.

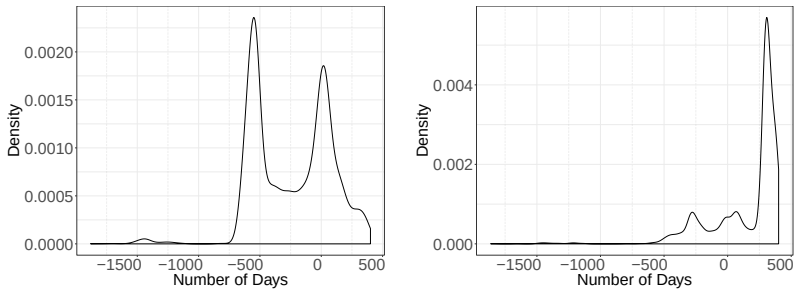
5.3 Reputation Analysis

In this section, we explain how we intersect publisher domains from DSP traffic with blacklists and malware domains to perform reputation analysis.

Public Blacklist Traffic Since 89.87% of the domains on the blacklists we collected do not have semantic information, we filter them to ensure they are bad publishers with high confidence. We want to be conservative about what we keep, so we choose the following filters. First, we obtained all the domains that appeared on the Alexa [11] top one million list for every day from 12/10/2014 to 3/24/2015. We excluded those consistent Alexa domains because they are unlikely to be malicious. Second, we excluded all domains under the ad server category of EasyList [1], because malware conducting impression fraud or click fraud can generate traffic that goes through ad servers. Lastly, we excluded a hand curated a whitelist of CDN effective second level domains (e2lds) and we excluded all fully qualified domain names that overlapped with these e2lds.

Observation 1: **0.27% publisher domains appeared in DSP traffic were blacklisted by the security community. They generated 1.8% of all bid requests daily.**

We observed 4,905,224 unique domains in the DSP traffic from 12/10/2014 to 3/24/2014. Among them, 13,324 (0.27%) domains were blacklisted some time between 12/9/2009 and 1/15/2016. Blacklisted domains were responsible for an average of 1.8% of all bid requests every day. Previous studies estimate nearly



(a) First seen dates (b) Last seen dates

Fig. 7: Density plot of first seen date date on PBL - first date seen from DSP (7a) and last seen date on PBL - last date seen from DSP (7b).

30% of bid requests are malicious [14, 16], which suggests this is only a fraction of the actual abuse. While there are many potential causes, such as referrer spoofing or lack of ad-abuse investigations, these findings show simply relying on blacklists from the security community is insufficient to study and combat abuse. While they are few, we investigated the potential to automatically detect these abusive domains.

Observation 2: **68.28% of blacklisted publisher domains were known to the security community before they appeared in DSP traffic.**

Figure 7a shows the density distribution for the difference of days between when a domain was first blacklisted and when it was seen in DSP traffic. The zero value in this case means that the domain name was blacklisted on the same day as it was seen in the ad exchanges. Similarly, a value of -500 means that the domain was blacklisted 500 days before it ever appeared in the datasets from the DSP. The plot shows that 68.28% (9,097) of all blacklist domains were known to the security community prior to they started requesting for ads in the DSP traffic. Moreover, 32.49% (4,329) of blacklisted publisher domains were labeled more than 535 days before they were seen in the DSP datasets. The peaks of the distribution reflects several blacklists update events. One event was a major update of 4,031 domains on 6/23/2013, which corresponds to the -535 days in Figure 7a. Another update event on 12/4/2014 was reflected around -6 days in the plot. Eighty domains were blacklisted on 1/15/2011, which makes up the small bump around -1500 days in the plot.

Figure 8 is a scatter plot of the first date a domain is blacklisted (x-axis) and its corresponding first seen date in the DSP (y-axis). The size of the point represents the number of domains in these dates. The points in the bottom side of the plot are large because this is the first date we had the DSP data. The vertical group of points represent domains being updated in the blacklist in the same day. We highlighted a few days when blacklisted domains from the DSP traffic were first labeled. The plot is more dense on the right side since 2013-06-23. We increased the number of blacklists to crawl from 3 to 7 on that day, which resulted

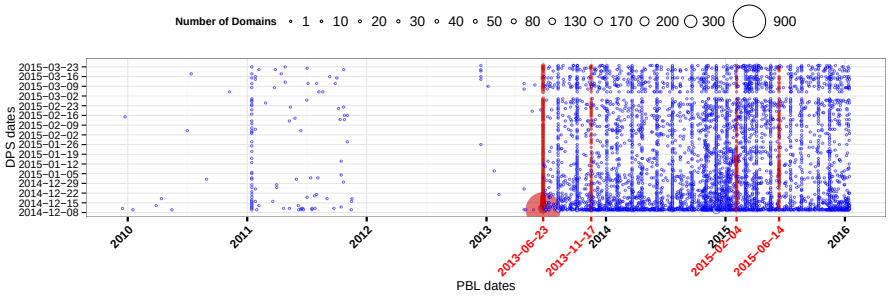


Fig. 8: Scatter plot of first date seen on PBL and first date seen from DSP for all DSP domains that were on PBL.

in more domain names in PBL dataset and more overlap with the DSP traffic from that point on. On 2013-11-17, the blacklists updated many domain names including `websearch.*.info` used by browser hijackers. On 2015-02-04, there were a lot of "update" domains used by pop-up ads added to the blacklists, e.g., `soft12.onlineupdatenow.com`. On 2015-06-14, the blacklists updated a group of algorithmically generated domains with sub domains `freempr#`.

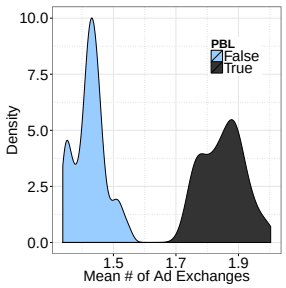
Observation 3: **Most (77.01%) blacklisted publisher domains remained on blacklists after they were last seen in DSP traffic.**

We would like to see whether the publisher domains remained on the blacklists after they were seen in the DSP. We plotted the density distribution for the number of days when a domain was last seen on blacklists minus when it last appeared in the DSP (Figure 7b). The distribution has shifted a lot towards the right part of the x-axis this time. Figure 7b shows that the majority (77.01%) of blacklisted domains were still on blacklists after they were seen in the DSP. A total of 14.06% (1,873) of them remained on blacklists more than a year after they were last seen in the DSP datasets. The peak of Figure 7b reflects the last date (1/15/2016) of our blacklist dataset. Overall 8,051 DSP domains belong to this peak in the plot.

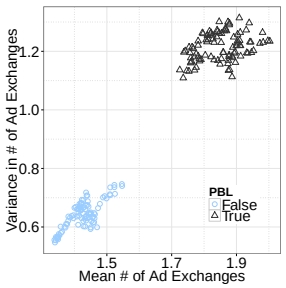
Observation 4: **Blacklisted publisher domains tend to use more ad exchanges and reach more hashed client IP addresses than those that have never been blacklisted.**

Each day, we separate the publisher domains into two groups: those that were seen in PBL (True) and not in PBL (False). For each group, we compute the average number of distinct ad exchanges and the number of hashed client IPs that a publisher domain was seen from, as well as the variance within the group. We visualize the results in Figure 9a to Figure 9d.

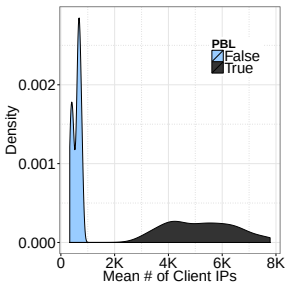
Figure 9a shows the density distributions of the daily average number of ad exchanges for the PBL group and non-PBL group across the entire DSP dataset. The PBL group were seen from an average of 1.7 to 2 ad exchanges, more than the non-PBL group. We perform a two-sample Kolmogorov-Smirnov



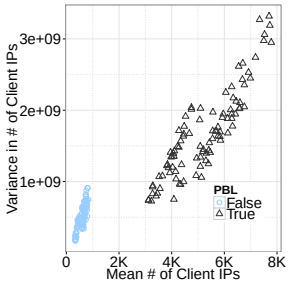
(a) Pbl: # of Ad Exchanges Density



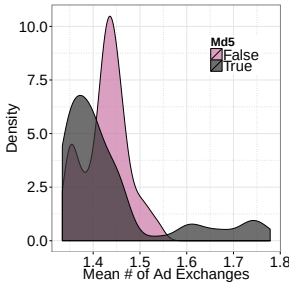
(b) Pbl: Mean, Variance for # of Ad Exchanges



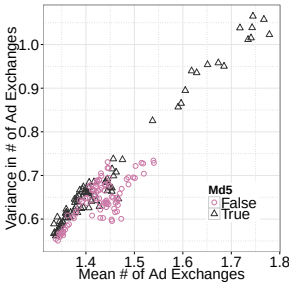
(c) Pbl: # of hashed IPs Density



(d) Pbl: Mean, Var for # of hashed IPs

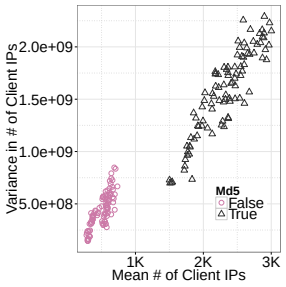


(e) Md5: # of Ad Exchanges Density

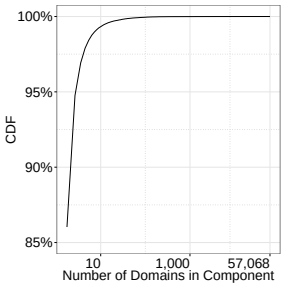


(f) Md5: Mean, Variance for # of Ad Exchanges

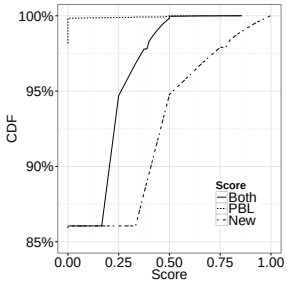
(g) Md5: # of hashed IPs Density



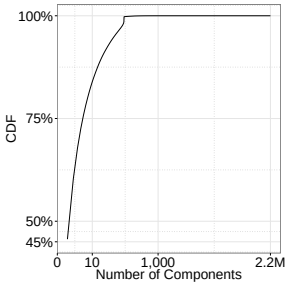
(h) Md5: Mean, Var for # of hashed IPs



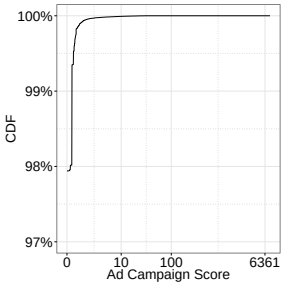
(i) Components of 12/10/2014



(j) Three scores on 12/10/2014



(k) Component sizes in ad campaigns



(l) Ad campaign scores

Fig. 9: 9a to 9d are PBL plots. 9e to 9h are Md5 plots. 9i to 9l are CDFs for number of publisher domains forming components of 12/10/2014 (9i), three scores for components seen on 12/10/2014 (9j), number of components in ad campaigns (9k) and ad campaign scores (9l).

test (K-S test) where the null hypothesis is that x=y, i.e., that the datasets are drawn from the same distribution. The K-S test demonstrates we can reject this null hypothesis ($p - value < 2.22 * 10^{-16}$). Therefore, the two distributions are significantly different. We also plot the mean and variance of the average ad exchange number for each group in Figure 9b. The figure shows that not only do non-PBL domains use fewer ad exchanges in general, the difference of the measure between non-PBL domains is small, as reflected by the variance. On the other hand, PBL domains have relatively higher variance among themselves.

Similarly, we plot the density distribution for number of average hashed client IP addresses in a day for the PBL and non-PBL groups (Figure 9c), as well as the mean and variance of the metric (Figure 9d). These figures show that PBL domains tend to be seen from more hashed client IPs than non-PBL domains. Since the majority of the content on the web is in the unpopular "long tail", only a few hashed client IPs visit any non-PBL domain in general, and the variance of number of clients is low (Figure 9d). In contrast, PBL domains seen in the RTB process aim to make money, and thus spread to as many hosts as possible.

Malware Traffic Domains queried by malware are another type of threat information commonly used by the security community. We filtered the malware domains using the same three methods as in the PBL case. Within 4,905,224 unique domains from the DSP traffic, 134,262(2.74%) were queried by malware samples collected over five years. *There are ten times more publisher domains queried by malware than from those on blacklists.* Similarly, we can separate the publisher domains into two groups: malware domain group (Md5 True) and non-malware domain group (Md5 False). We computed the average daily number of ad exchanges and hashed client IP addresses for each day in the DSP traffic.

Observation 5: **Malware domains have different behavior than blacklisted domains. That is, malware domains were observed to employ similar number of ad exchanges to non-malware domains, however, with a higher number of hashed client IP addresses.**

Figure 9e to Figure 9h show the measurement results. We observe bimodal distributions of malware vs. non-malware domains in Figure 9e and Figure 9g. Figure 9e and Figure 9f show that publisher domains queried by malware tend to use a similar number of ad exchanges. In addition, the distributions between malware domains and non-malware domains overlapped much more than when we compared PBL group with non-PBL group. Therefore, the number of ad exchanges is not a distinguishing attribute for the MD5 group. On the other hand, DSP domains queried by malware were still seen from a larger group of hashed client IP addresses, compared to the rest of domains never queried by malware. Malware domains that interact with ad ecosystem are relatively more popular than non-malware domains.

Malware query non-malicious domains for various reasons, and only a few of the domains are fraudulent publishers. Recall that when malware interacts with the ad ecosystem from the client side (Figure 1), there may be syndicated publishers, or benign ad servers contacted by the malware, in order to reach ad

exchanges. Despite our filtering efforts, it is likely that there are still numerous benign domains in the malware domain set. Additionally, domains could remain on blacklists after they become inactive or parked, which results in false positives when using blacklists. These findings all point to the need for better ad-abuse ground truth datasets.

6 Infrastructure Tracking

In this section, we show that traditional DNS infrastructure features can be used to extend the ground truth set, discover new ad abuse cases and track the threat evolution over time. This can be used by any entity in the ad ecosystem with visibility of bidding requests to track advertising campaign infrastructure—focusing on those that are likely to be malicious in intent. While we acknowledge that the word "campaign" has an overloaded meaning, we define it in the following way and only in the context of ad abuse: *a campaign will be defined as the set of domain names that can be linked together over time based on their IP infrastructure properties.*

At a high level, we construct graphs of the relationship between the domain name of the ad publisher and the infrastructure the domain name uses. By building and merging these graphs over time, we can track the infrastructure and focus on those campaigns that may be malicious, e.g., domains known to have been blacklisted, queried by malware, or have never been seen before. We present case studies based on this process in Section 7.

6.1 Constructing Infrastructure Graphs

An *infrastructure graph* is an undirected graph G, defined by its set of vertices V and edges E. A *disconnected* graph is made up of multiple *components* or subgraphs with no adjacent edges between them. These components correspond to advertising campaigns that are tracked over time. Vertices in infrastructure graphs are domain names or the RDATA the domain names resolve to. RDATA can be an IPv4/IPv6 address (A/AAAA), a canonical name (CNAME), or a nameserver (NS). Two vertices are adjacent if and only if exactly one is a domain name, and the domain name resolved to the RDATA of one of the aforementioned query types (A/AAAA/CNAME/NS) during time t when the domain name appeared as a publisher for a bid request.

A Demand Side Platform provider (DSP) can build infrastructure graphs by performing the following steps. First, the DSP collects all publisher domain names D_p from the bid requests seen on day t. Second, the DSP resolves all domain names $d \in D_p$, which results in zero or more domain name and IP address tuples. More formally, resolving d will yield $[(d, rdata_0), \cdots, (d, rdata_N)]$ if d resolves to N different IPs, CNAMEs, or NSes on day t. Each of these tuples corresponds to an edge in our graph G. Finally, after G is built for day t, G is decomposed into its *connected components* C, where each component $c \in C$ is ranked and tracked over time as a specific ad campaign. While we experimented

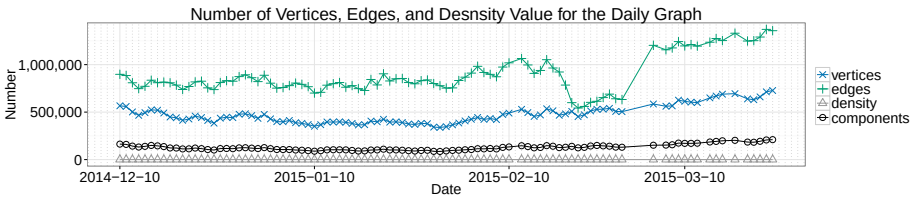


Fig. 10: Number of vertices, edges and density values for the graph every day.

with more sophisticated community discovery or spectral methods, the benefits gained were disproportional to the add-on complexity. Thus, we decided to select the simplest and most straightforward way to mine the graph for campaigns.

Since the DSP bidding request traffic did not include DNS resolution information, we chose to correlate that with the DNS dataset obtained from a passive DNS database from a North American ISP (Table 1). By combining the DNS resolution seen in the same day in the ISP with the publisher domains from the bidding request traffic, we were able to construct daily infrastructure graphs. Next, we discuss how we analyze the produced graphs.

Graph Analysis We study the infrastructure graphs using some basic graph analysis metrics. Specifically, we first analyze overall graph properties including vertices, edges and density measures. Then, we examine the connected components of the graphs every day and over time. These analytics help us understand the infrastructure of the publisher domains, and give us insights about how to rank components based on how suspicious they are and track them over time.

First, we discuss three properties of daily infrastructure graphs. Figure 10 shows three statistics for graphs generated every day: number of vertices (V), number of edges (E), and the density measure. We use the following formula to compute the *graph density* D:

$$D = \frac{2E}{V(V-1)} \tag{1}$$

On average, there are 472 thousand vertices, and 883 thousand edges every day. The graphs are extremely sparse and the daily density is only $8.35 * 10^{-6}$. In fact, the majority of the edges only connect two vertices. There are 566,744 vertices on 12/10/2014, and it dropped to 342,426 (by 39.58%) on 1/29/2015. Then the number of vertices slowly increased to 727,501 on 3/24/2015. Since vertices include publisher domains and DNS resolution data, the change in the number of vertices over time is largely consistent with the observation of how the number of daily publisher domains changed (Figure 4). On the other hand, the change in the number of edges per day is different. The number of daily edges decreased since 2/17/2015, and dropped to the lowest number 542,945 on 2/21/2015, before it jumped up to 1,203,202 on 3/5/2015. Through manual

analysis, we concluded that this was not caused by any single domain name. There were fewer resolved data per domain in general in these days.

Second, we study properties of connected components in the infrastructure graphs. Figure 10 shows the number of connected components over time that were in the daily infrastructure graphs. On average, there are 127,513 connected components in a day. Figure 9i demonstrates that the daily infrastructure graph is highly disconnected. The cumulative distribution for the size of the components in a day follows the Zipf's law. For instance, CDF in 12/10/2010 shows that 86% of connected components have only one publisher domain in it. Fewer than 0.7% components have more than ten publisher domains.

6.2 Identifying Suspicious Components

The number of graph components based on the results from Section 6.1 can be hundreds of thousands in a day (Figure 10), which is likely too many for manual analysis. However, the measurement from Section 5 suggests we can prioritize components that are likely to be interesting from a security perspective. We know publisher domain names differ in behavior when they are known to appear on blacklists. Conversely the subset of malware domains seen in DSP are very noisy, and thus it is not a good metric to use for prioritizing components. We also hypothesize that never-before-seen domains deserve close scrutiny as they may represent infrastructure changing to avoid detection. The question remains if these are indicative of true malicious behavior. To find out, we rank publisher components by their domain names, specifically, if they are on blacklists, if the domains have never been seen before and a combination of these two measures.

For each publisher component $c \in C$ we compute two values β_c and ν_c that correspond to the proportion of domains in c that appear on blacklists, and are under brand new from the perspective of the DSP, respectively. Intuitively, the first one indicates an association with known malicious activity, and the last suggests the potential threat may have just begun. Specifically, the way we compute each value of a component is smoothed.

$$\beta_c = \frac{\#\ of\ blacklisted\ publisher\ domains - 1}{Total\ \#\ of\ publisher\ domains} \tag{2}$$

$$\nu_c = \frac{\#\ of\ brand\ new\ publisher\ domains - 1}{Total\ \#\ of\ publisher\ domains} \tag{3}$$

We offset the numerator count by one based on results of the infrastructure graph analysis from Section 6.1. Since the majority of components have only one publisher domain name in it, they are isolated singletons and do not provide any information to other unlabeled domains from infrastructure point of view. We prefer not to prioritize these singletons among all components even if they are already blacklisted or brand new. Equation 2 and Equation 3 give singleton components both zero values. Moreover, we judge whether a domain name is "brand new" using the effective second-level domains (e2ld) according to public suffix list [9]. An e2ld is the smallest registrable unit of a domain name and two

domains under an e2ld are likely operated by the same individual. Therefore, a new domain under a new e2ld is more interesting to us.

After getting these two values β_c and ν_c, we also compute the linear combination of these: $\iota_c = \frac{1}{2}(\beta_c + \nu_c)$. Finally, we reversely sort the components in a day based on the ι_c score. Within a day, ι_c can range between 0 and 1. A component with higher ι_c will be prioritized over a component with lower ι_c for inspection. Figure 9j presents cumulative distributions of the proportion of pbl-related, never-before-seen domains and a linear combination of the two for a day per component. A total of 98% of the components have zero PBL score because they do not have any blacklisted domains, and 14% of the components have a score for having new domains. The final component score combining the two falls in between the two distributions.

6.3 Tracking Campaigns Over Time

Building infrastructure graphs for an individual day is useful, but tracking the ad campaigns over time will yield more comprehensive coverage of ad campaigns, as well as advanced warning of potentially malicious ones. First, if an ad campaign is determined to be malicious, tracking them over time through small infrastructure changes will enable more comprehensive blacklists to be built. Second, if a tracked ad campaign is known to be malicious, newly added infrastructure can be more pro-actively blacklisted. Finally, tracking infrastructure over time allows us to build ground truth to eventually model malicious and benign advertising campaign infrastructure. In our future work we plan to experiment with predicting fraudulent publishers.

To unify ad campaigns across multiple infrastructure graphs, we simply join ad campaigns that share IP addresses, canonical names, and name servers that are the same. This allows us to not only construct graphs within days, but also across time. We will show that this simple tracking method works well in practice. While on average there are 127K connected components every day, only 10K of them form new ad campaigns. A DSP can choose to only go through top-ranked new components if there is limited time available for threat analysts.

ι_{ad} is used to sort advertising campaigns to identify case studies. It is calculated by adding up all the interesting scores of individual components ι_c belonging to that campaign. After we sort the ad campaigns by ι_c, we then examine the distribution of the interesting scores and number of components in the campaigns. Figure 9l shows the cumulative distribution of ad campaign scores. Also, Figure 9k shows the CDF of the number of components in an ad campaign. Overall 99.99% ad campaigns have fewer than 1,000 components. The ad campaign with the largest number of components (2.2 million in Figure 9k) has the highest ad campaign score. Domains in this campaign resolved to several parking, and sinkholing IP addresses, as well as common names servers like GoDaddy. This is the reason that this noisy campaign is not representative of maliciousness or freshness of the domains. Starting from the second ad campaign, the interesting score indicate suspicious activities in the ad exchanges. We now describe the case studies this measure uncovers in Section 7.

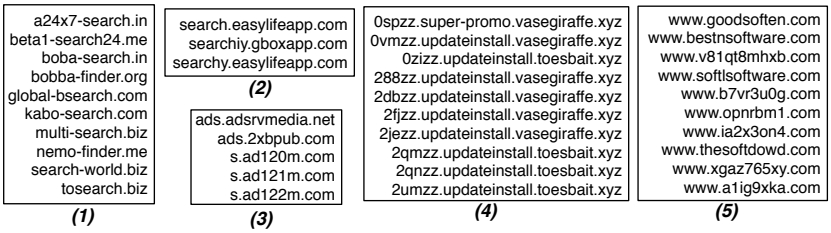


Fig. 11: Publisher domain examples.

7 Case Studies

Among the campaigns with highest (top 0.1%) interesting scores, we found new cases including Potentially Unwanted Programs (PUP), algorithm generated domains and malware sites.

7.1 Case 1: PUP

Among advertising campaigns with the highest interesting scores, one category of publisher domains are generated by Potentially Unwanted Programs (PUP). For example, domains in Figure 11 (1) to (5).

A VirusTotal report [35] suggests a machine communicating with domain names in Figure 11 (1) (ι_{ad} ranked the 3rd highest) is likely infected with a trojan known as LEMIR or Win32.BKClient by the AV industry. The malware has many capabilities including changing default search engines to generate revenue, disabling Windows AV, Firewall and Security Center notifications, and can drop additional malicious binaries [33]. Similarly, ad campaigns with 2^{nd} and 4^{th} highest ι_{ad} (Figure 11 (2) (3)) are generated by ad injections of certain browser extension. Different malware families communicate with domains in Figure 11 (3) including Win.Trojan.Symmi [36]. These publisher domains may not be malicious, but they are strongly associated with monetization behavior of malware. These are interesting cases as traditional malware are involved in an area where we would expect to see only adware or "potentially unwanted programs." This shows that malware uses advertising fraud to monetize infections and malware can also be identified from the vantage point of a DSP.

In addition, several Pop-up Ads campaigns exhibit high level of agility similar to traditional malware. The ad campaign ranked $1,184^{th}$ (Figure 11 (4)) uses domain fluxing, likely to avoid browser extension detection systems. In total, we observed more than 26,000 unique domain names from this campaign in three months of DSP traffic. Moreover, the ad campaign in Figure 11 (5) not only uses domain fluxing, it also uses the Amazon EC2 cloud to further decrease the chance of detection. Each of these domains resolved into an EC2 cloud domain representing a unique Virtual Machine (VM), when active. The VM domains also change according to the domains that point to them. This shows that miscreants are constantly employing fresh VMs to perform ad fraud. Since traditional

Fig. 12: Malware site example.

detection systems often use reputation of IP addresses of domains and URLs, using cloud machines makes this campaign harder to be detected.

7.2 Case 2: Algorithm Generated Domains

Figure 12 (1) (2) shows two ad campaigns of algorithm generated domains we found in the DSP traffic (ranked 142^{th} and 183^{th}), containing at least 195 domains. None of the domains were blacklisted, but a high percentage of brand new domains results in a high score. A new group of domains appear everyday, pointing to the same IP address. These publisher domains are suspicious. Although no open threat analysis evidence is available to date, it is reasonable to assume that anything that changes so often must be trying to evade a detection process. With infrastructure tracking, ad exchanges or DSP can keep a close eye on such campaigns to proactively deal with potential ad abuse.

7.3 Case 3: Malware Site

Figure 12 (3) shows a group of malware site domains (ranked $1,484^{th}$ campaign) seen from DSP traffic, none of which appeared on blacklists. A Virustotal report [37] shows that the IP address these domains resolved to, had other similar domains pointing to it during the week ending on 3/24/2015. Related URLs were detected as malware sites by several URL scanners from the AV industry. This group uses domain fluxing with both the second level domain zone, and the child labels. We saw other groups of domains tracked separately, with similar domain name patterns, and short lifetime. However, they were not grouped into one big campaign, because different groups were using different IP addresses. In other words, this campaign uses both domain fluxing and IP address fluxing. Since we only used exact the same IP address match to form a campaign, we will need other information to further analyze campaigns like this.

8 Related Work

Previous research has studied behavior of click bots [17, 18, 26]. The bots mimic human behavior by generating fake search queries and adding jitters to click

delay. More advanced bots hijacked users' original clicks and replaced the ads [12, 13,18,28]. The ZeroAccess botnet cost advertisers $100,000 per day [28] and the TDSS/TDL4 botnet cost advertisers at least $346 million in total. Ad fraud detection work mainly focused on click fraud [19, 25, 32].

Impression fraud is harder to detect than click fraud. Springborn et al. [29] studied pay-per-view networks that generated fraudulent impressions from invisible iFrames and caused advertisers millions of dollars lost. Advertisers can purchase *bluff ads* to measure ad abuse [18] and compare charged impressions with valid impressions. The adware and ad injection problem has been systematically studied by static and dynamic analysis of web browser extensions [21, 31, 38]. From within the ad ecosystem, Stone-Gross et al. [30] used ad hoc methods to study specific attacks faced by ad exchanges, including referrer spoofing and cookie replay attacks. Google also documented what they consider to be invalid traffic in [34] but did not disclose the details of their traffic filters.

9 Conclusion

In this study, we measured ad abuse from the perspective of a Demand Side Platform (DSP). We found that traditional sources of low reputation, such as public blacklists and malware traces, greatly underestimate ad-abuse, which highlight the need to build lists catered towards ad-abuse. The good news, however, is malicious publishers that participate in ad-abuse can likely be modeled at the DSP level based on their behavioral characteristics. Finally, malicious campaigns can be tracked using graph analysis and simple heuristics, allowing DSPs to track suspicious infrastructure.

Acknowledgements. We would like to thank TAPAD and in particular their CTO, Dag Liodden, for his invaluable help throughout this project. 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. 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. EasyList. `https://easylist-downloads.adblockplus.org/easylist.txt`.
2. Hphosts List. `http://hosts-file.net/?s=Download`.
3. I.T. Mate List. `http://vurldissect.co.uk/daily.asp`.
4. Malc0de Database. `http://malc0de.com/bl/BOOT`.
5. Malware Domain List. `https://www.malwaredomainlist.com/`.
6. PassiveTotal: RiskIQ. `https://www.passivetotal.org/`.
7. sagadc.org list. `http://dns-bh.sagadc.org/domains.txt`.

8. SANS ISC Feeds. https://isc.sans.edu/feeds/.
9. Mozilla Public Suffix List. https://publicsuffix.org/list/, 2015.
10. Advertising Age. Ad Fraud Will Cost $7.2 Billion in 2016, ANA Says, Up Nearly $1 Billion. http://bit.ly/1Qe21C2.
11. Alexa. The web information company. http://www.alexa.com/, 2007.
12. S. A. Alrwais, A. Gerber, C. W. Dunn, O. Spatscheck, M. Gupta, and E. Osterweil. Dissecting ghost clicks: ad fraud via misdirected human clicks. In *Proceedings of the 28th Annual Computer Security Applications Conference*. ACM, 2012.
13. M. Antonakakis, J. Demar, K. Stevens, and D. Dagon. Unveiling the network criminal infrastructure of tdss/tdl4 dgav14: A case study on a new tdss/tdl4 variant. Technical Report, Damballa Inc.,Georgia Institute of Technology (GTISC), 2012.
14. Association of National Advertisers. The Bot Baseline: Fraud in Digital Advertising. http://bit.ly/1PKe769.
15. Y. Chen, P. Kintis, M. Antonakakis, Y. Nadji, D. Dagon, W. Lee, and M. Farrell. Financial lower bounds of online advertising abuse. In *International conference on Detection of Intrusions and Malware, and Vulnerability Assessment*, 2016.
16. ClickZ. Fake Display Ad Impressions Comprise 30% of All Online Traffic [Study]. http://bit.ly/2e3HdCZ.
17. N. Daswani and M. Stoppelman. The anatomy of Clickbot.A. In *the First Workshop on Hot Topics in Understanding Botnets*. USENIX Association, 2007.
18. V. Dave, S. Guha, and Y. Zhang. Measuring and fingerprinting click-spam in ad networks. In *Proceedings of the ACM SIGCOMM 2012 conference on Applications, technologies, architectures, and protocols for computer communication*.
19. V. Dave, S. Guha, and Y. Zhang. Viceroi: catching click-spam in search ad networks. In *2013 ACM SIGSAC conference on Computer & communications security*.
20. Department of Homeland Security. Trusted Cyber Risk Research Data Sharing. https://www.dhs.gov/csd-impact.
21. A. Kapravelos, C. Grier, N. Chachra, C. Kruegel, G. Vigna, and V. Paxson. Hulk: Eliciting malicious behavior in browser extensions. In *23rd USENIX Security Symposium (USENIX Security)*, 2014.
22. Malware Tips. How to remove Websearch.searc-hall.info. http://bit.ly/2e9qyKw.
23. Malware Tips. Remove Sl.now-update-check.com virus. http://bit.ly/2dm1LWp.
24. W. Meng, R. Duan, and W. Lee. DNS Changer Remediation Study. In *M3AAWG 27th General Meeting*, 2013.
25. A. Metwally, D. Agrawal, and A. El Abbadi. Detectives: detecting coalition hit inflation attacks in advertising networks streams. In *Proceedings of the 16th international conference on World Wide Web*, pages 241–250. ACM, 2007.
26. B. Miller, P. Pearce, C. Grier, C. Kreibich, and V. Paxson. What's clicking what? techniques and innovations of today's clickbots. In *Detection of Intrusions and Malware, and Vulnerability Assessment*. 2011.
27. openrtb.info. OpenRTB: Documentation and Issue tracking for the OpenRTB Project. http://openrtb.github.io/OpenRTB/, 2014.
28. P. Pearce, V. Dave, C. Grier, K. Levchenko, S. Guha, D. McCoy, V. Paxson, S. Savage, and G. M. Voelker. Characterizing large-scale click fraud in zeroaccess. In *2014 ACM SIGSAC Conference on Computer and Communications Security*.
29. K. Springborn and P. Barford. Impression fraud in online advertising via pay-per-view networks. In *Proceedings of the 22nd USENIX Security Symposium*, 2013.
30. B. Stone-Gross, R. Stevens, A. Zarras, R. Kemmerer, C. Kruegel, and G. Vigna. Understanding Fraudulent Activities in Online Ad Exchanges. In *Proceedings of the 2011 ACM SIGCOMM conference on Internet measurement conference*.

31. K. Thomas, E. Bursztein, C. Grier, G. Ho, N. Jagpal, A. Kapravelos, D. McCoy, A. Nappa, V. Paxson, P. Pearce, et al. Ad injection at scale: Assessing deceptive advertisement modifications. In *2015 IEEE Symposium on Security and Privacy*.
32. T. Tian, J. Zhu, F. Xia, X. Zhuang, and T. Zhang. Crowd fraud detection in internet advertising. In *Proceedings of the 24th International Conference on World Wide Web*, pages 1100–1110. ACM, 2015.
33. TrendMicro, Inc. Threat Encyclopedia: TROJ_LEMIR.CS. `https://goo.gl/8ryRjK`, 2012.
34. A. Tuzhilin. The Lane's Gift v. Google Report (2006).
35. VirusTotal. Antivirus scan. `https://goo.gl/jU0b0b`, 2014.
36. VirusTotal. Antivirus scan. `https://goo.gl/s97XI5`, 2015.
37. VirusTotal. IP address information. `https://goo.gl/ifLvT5`, 2015.
38. X. Xing, W. Meng, B. Lee, U. Weinsberg, A. Sheth, R. Perdisci, and W. Lee. Understanding malvertising through ad-injecting browser extensions. In *Proceedings of the 24th International Conference on World Wide Web*, 2015.
39. Zeus Tracker. Zeus IP & domain name block list. `https://zeustracker.abuse.ch`.