Follow the traffic: stopping click fraud by disrupting the value chain

Matthieu Faou*, Antoine Lemay*, David Décary-Hétu†, Joan Calvet‡, François Labrèche*,
Militza Jean*, Benoit Dupont† and José M. Fernandez*
*École Polytechnique de Montréal
†Université de Montréal
‡ESET

***Abstract*—Advertising fraud, particularly click fraud, is a growing concern for the online advertising industry. The use of click bots, malware that automatically clicks on ads to generate fraudulent traffic, has steadily increased over the last years. While the security industry has focused on detecting and removing malicious binaries associated with click bots, a better understanding of how fraudsters operate within the ad ecosystem is needed to be able to disrupt it efficiently.**

This paper provides a detailed dissection of the advertising fraud scheme employed by Boaxxe, a malware specializing in click fraud. By monitoring its activities during a 7-month longitudinal study, we were able to create of map of the actors involved in the ecosystem enabling this fraudulent activity. We then applied a Social Network Analysis (SNA) technique to identify the key actors of this ecosystem that could be effectively influenced in order to maximize disruption of click-fraud monetization. The results show that it would be possible to efficiently disrupt the ability of click-fraud traffic to enter the legitimate market by pressuring a limited number of these actors. We assert that this approach would produce better long term effects than the use of take downs as it renders the ecosystem unusable for monetization.

I. Introduction

The development of the Internet enabled a wealth of content to become readily accessible. A large volume of this content is offered for free. This is true even for content that we used to pay for, such as newspapers. Naturally, content creators need to make up for the absence of income by finding a new revenue stream. This revenue stream is Internet advertisement. By showing ads to their visitors, and having click on those ads, content creators are able to convert traffic into a revenue stream. This business model is now a dominant force on the Internet, with the size of the market in 2014 estimated at 59.6 billion dollars in the US alone [18], and 159.8 billions dollars worldwide [29].

However, criminals can also abuse this system to monetize computers infected with malware. By distributing specialized ad fraud payloads to these machines, fraudsters can generate a large number of requests that resemble those produced by humans. They can then get revenue from this fake traffic without needing to invest in content creation. In fact, the Association of National Advertisers estimated that, in 2015, publicity fraud will cost more than 6 billion dollars to advertisers worldwide [36], representing close to 4% of total global publicity revenue. The prevalence of this problem undermines the business model that underpins most free services on the Internet today. Thus, it is imperative to find ways to better understand and address this problem.

One technique that can be used to combat this form of fraud is the disruption of the so-called *value chain*, i.e. the links between fraudulent actors and legitimate businesses through which fraudsters acquire wealth. One example of the successful use of this method was the campaign to shut down payment processors used for scareware [22]. In that case, the monetization scheme was the distribution of a fake anti-virus (fake AV). The user was informed that his computer was infected and was then offered a fake AV product to clean the infection. The "product" had to be purchased through a credit card transaction. The fact that these credit card transactions could be linked to particular payment processors, allowed credit card companies to stop this kind of transactions. But in the case of publicity fraud, this is not so easy. First, money changes hands several times before reaching the fraudsters. Furthermore, there is no centralized database containing all of these transactions that could be analyzed to understand where the money going to fraudsters comes from. How can we then build a global picture of the business relationships between actors involved in ad fraud, willingly or unwillingly, in order to identify such choke points where disruptive pressure can be applied?

One possible method is to first reconstruct the traffic redirection chains involved in advertising to map the corresponding value chains. By following a series of redirections taken by an automated click-fraud module from the infected computer to the advertiser, it is possible to get a glimpse of how traffic changes hands between the different actors involved in click-fraud. Every time traffic is transfered between actors, there is a corresponding economic transaction. Much like police officers doing surveillance on drug dealers, by observing enough transactions, it is in principle possible to reconstruct the entire network of actors involved in such illicit activities. The network can then be analyzed to find optimal targets for disruption.

In this paper, we present the results of our 7-month longitudinal study on the automated click-fraud network associated with the Boaxxe malware. We reconstruct a map of the actors involved in Boaxxe's fraud network by analyzing and aggregating the *redirection chains* gathered from observation

of the network activity of machines infected this malware. We describe the structure of this click-fraud ecosystem and identify potential critical targets in it. The paper starts by providing some background about Internet publicity and automated click-fraud. It follows with a description of our data collection methods. We then present the results of our longitudinal study and the disruption possibilities identified. We also discuss how the methods and findings presented in this paper could be applied to design and implement more effective and generic anti-click fraud policies and strategies. Finally, we offer a brief conclusion.

II. Background

In this section, we introduce basic notions related to the legitimate Internet advertising ecosystem and we present some of the techniques used to defraud this market.

A. Online advertising

To fully understand ad fraud, it is critical to know the basics of online advertising.

Advertiser: A person or a company that wants to promote its products or services. This entity pays to *display* ads on other web sites to attract visitors to them. It may also pay per traffic redirected to its web site by other web sites as a result of a human action, i.e. a *click* on the corresponding ad.

Publisher: A person or a company running a web site that displays advertisements to its visitors. This entity earns money by showing ads and by having users click on these ads.

Ad network: A person or a company that buys and sells ads or visitors. It buys and sells traffic in bulk through pre-established contracts, or through *ad exchanges* which are automated auction markets where traffic is bought and sold instantly. In other words, it is an intermediary between advertisers and publishers. Ideally, an ad network aims to match the right ad with the right visitor in order to serve the needs of its ultimate client, the advertiser. This is done, for example, by matching the advertiser's requirements for visitors with the user profiles constructed from browsing information, e.g. browsing history, search history, Web site cookies, etc. Not all traffic brokers operate their own infrastructure. Some are only marketing companies and rely on Solution-as-a-Service (SaaS) providers to manage their ad serving infrastructure.

Compensation: Ad networks generally propose different types of compensation. The three main types are *Cost Per Mille*[1] (CPM), *Cost Per Click* (CPC) and *Cost Per Action* (CPA). The cheapest is CPM, where each displayed ad is typically compensated with a few tenths of a cent. This is relatively low price is due in part to the fact that there is little guarantee that visitors are interested in the ad and will take further revenue-production actions for the advertiser. On the other hand, the fact that a visitor clicks on the ad (CPC) provides better chances of revenue and accordingly clicks are typically more expensive. Prices wary widely depending on the advertiser category, but mean prices are in the dollar range [37]. Finally, clicks that can be linked to an actual revenue-generating action by the user, i.e. CPA, are compensated the most.

[1]Cost per thousand views. This terminology comes from traditional advertising.

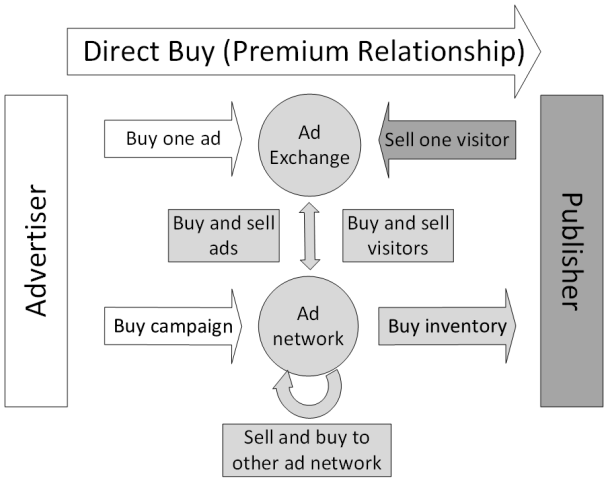

Fig. 1. Advertisement ecosystem

When an advertiser wants to attract traffic for its web site, its ad network will promote the web site by displaying *creatives*, such as text banners, pictures or ad videos, on other web sites. For the advertiser, this is the equivalent of buying visitors. The ad network can also take action so that the advertiser web site appears more prominently in search engine results, i.e. "sponsored" sites. In other words, search engines can also be considered publishers and are compensated for bringing traffic to the advertiser. This property is exploited in the click-fraud strategy studied in this paper. Figure 1 provides a pictorial description of the advertisement ecosystem.

To maximize revenue, ad networks often resort to *arbitrage*. At first, ad networks bought traffic only for their own advertiser customers. However, many of them now also buy traffic speculatively in order to resell it at a profit to other ad networks, either in bulk or through ad markets. This practice has effectively and the establishment of ad markets has effectively transformed Internet publicity traffic into a tradeable *commodity*. Even though publicity is an intrinsically perishable commodity, the advent of very low latency ad markets allow the same traffic (ad display or click) to be sold and re-sold many times before it lands on the advertiser's web site. As a result, the value chain between the published and the advertiser for the same piece of traffic can become quite long and complex.

One of the conditions that makes arbitrage possible and profitable is that the price of traffic varies significantly in time and according to user profile. The price of a piece of traffic can vary over time due to changes in supply (e.g. time of day, season, peaks in Internet usage) or demand (e.g. advertising campaigns), and of course the time-effects of a speculative high-frequency market itself. In addition, prices vary depending on the origin of the traffic, i.e. user location and demographics. For example, a luxury watch vendor is

TABLE I
EXAMPLE OF A REDIRECTION CHAIN. THE FIRST DOMAIN IS THE PUBLISHER AND THE LAST DOMAIN IS THE ADVERTISER. THE DOMAINS IN BETWEEN ARE THOSE OF THE INTERMEDIARY AD NETWORKS. THE REFERER FIELD IS NOT CHANGED BY HTTP 300'S REDIRECTIONS.

Position	Request	Redirection type	Referer field
1	**web-find.org/clk2?d=w4NK8...**	**HTTP 200**	/
2	web-find.org/r?q=kungfu4less&subid=...	HTTP 302	**web-find.org/clk2?d=w4NK8...**
3	**web-find.org/search?q=kungfu4less&subid=...**	**HTTP 200**	*- (unchanged)*
4	web-find.org/click?q=kungfu4less&subid=...	HTTP 302	**web-find.org/search?q=kungfu4less&subid=...**
5	88.214.241.236/click?sid=eef15...	HTTP 301	-
6	**207.244.71.165/redirect_js.php?ht_domain=web-find.org...**	**HTTP 200**	-
7	207.244.71.165/onclick.php?ht_domain=web-find.org...	HTTP 302	**207.244.71.165/redirect_js.php?ht_domain=web-find.org...**
8	207.244.71.165/local_bidding/onclick.php?affid=...	HTTP 302	-
9	adupmediaxml.com/bid_redirect.php?id_camp=...	HTTP 302	-
10	adupmediaxml.com/header_redirect.php?id_camp=...	HTTP 302	-
11	www.entrepreneur.com/topic/youve-arrived	-	-

more interested in visitors looking for watches and having high income and is willing to pay premium prices for these visitors. Similarly, it might be willing to pay higher prices one of its newer watches has been launched. These fluctuations in price create the opportunity for arbitrage, which in turns creates long value chains. As we will see, these longer value chains provide increased opportunity for fraudulent click to hide amongst legitimate traffic.

For search engines (SE), another monetization avenue is open through *syndication*[2]. In general, syndication is the process through which a publisher integrates external content from a *syndicator* onto its web page, e.g. through an API. In the world of Internet publicity, ad networks often act as syndicators providing ads to publishers. A particular example is that of a publisher operating a search engine, where the ads are sponsored links integrated in the search results through syndication, a process called *search-engine syndication*. Thus, when the user clicks on the sponsored link on the syndicated SE, he will be redirected to the syndicator's Web site before reaching the advertiser's Web page; this is necessary so that the SE can be credited accordingly.

This not only true for syndicated SE, but also for all publicity traffic. Every time that an ad network acquires traffic, the visitor is redirected to that ad network's Web site so that: 1) a decision can be made as to whom to sell the traffic to (an ad network or a publisher), and 2) construct and send an HTTP request to the buying party's Web site that adequately identifies the ad network's account. If the traffic is subsequently resold by the purchasing ad network, a similar process of redirection will be repeated, until the traffic reaches the advertiser. We call the corresponding sequence of HTTP requests and redirections the *redirection chain*. These redirections can take multiple forms, including through ordinary HTTP 300's redirect codes, but also through JavaScript or the HTML meta tag. To illustrate this process, an example of a redirection chain is presented in Table I. Some of these redirections are within an ad network's own site, but those with changing domain name or IP address typically correspond to a purchase and sale of traffic. In principle, these redirection chains can be reconstructed from the network traffic traces captured on the user's computer.

Note that in the CPM model, there is no guarantee that all intermediaries will be visible in the redirection chain. It would be theoretically possible for ad networks to have previous agreements where displays intermediaries are compensated without traffic transiting through their servers, for example by compensating them by percentage commission on CPM. However, we believe this is not the case in the CPC model because clicks are remunerated when an actual user clicks on a creative. The higher price of clicks and the unpredictability of supply (i.e. when users will click, what types of users will click) make it much less viable to have pre-negotiated contracts at fixed prices or percentages. Under this hypothesis, the redirection chain should contain most if not all of the intermediaries in the value chain between the publisher and the advertiser. It is an important assumption of our work, on which our results rely, that by looking at the network traffic generated by the web-browsing client machine, we can create a reasonable proxy of the value chain of the advertisement market.

B. Ad fraud

Knowing the volume of the online ad market, it is not surprising that it has become a prime target of fraudsters. Several techniques exist to defraud both advertisers and publishers. In this paper we limit ourselves to click fraud, i.e. the automated generation of fake clicks on ads to generate fraudulent revenue.

In this kind of fraud, the primary victims are the advertisers. They are buying clicks to increase audience and brand recognition. In turn, they expect this increased visibility to translate into increased revenue, through an increase in sales for example. However, that is only true if the traffic is from *bona fide* interested human visitors. If the visitors are scripts running on infected bots, little profit will be obtained from the clicks the advertiser paid for. Nonetheless, depending on the business model of the advertiser, it may be possible for the advertiser to shift the cost of the fraud elsewhere. If the advertiser that bought fraudulent traffic is *also* a publisher, it can still display its ads to the fraudulent visitor, pushing (partially) the costs to the advertisers paying for that ad space. In that sense, if the advertiser/publisher is able to perform arbitrage between the value of its ad space and the cost of

[2]This terminology is inherited from the world of electronic broadcast media.

buying traffic, it may even profit from click fraud. As an example of this, Bloomberg reported in an article published in September 2015 [15] that the Bonnier group, a bicentenary Swedish media company that recently launched several web sites, was buying botnet traffic to increase its audience and its own advertising revenue. In this paper, we will consider this odd phenomenon to be out of scope.

As for the ad networks, it is interesting to note that they are not always victims either. In fact, they can earn money for each click sold, as long as they receive more revenue for the click sold than for its purchased, and this even if the click is fake. The exception is when the fraudulent click is detected by the downstream ad network (or by the publisher); in that case, the ad network may not be compensated for a click it actually paid for. Thus, unscrupulous ad networks can be motivated to accept as much fraudulent traffic as possible without triggering fraud detection algorithms, such as was shown in the case of Yahoo in 2009 [21] who was forced to settle in a lawsuit involving click fraud transiting through their ad network services. Nonetheless, Mungamuru *et al.* [25] demonstrated that ad networks could actually benefit from aggressively fighting fraud. They argue that ad networks filtering fraudulent clicks most aggressively will have a competitive advantage, which could result into an increased market share. The rationale behind this conclusion is that the short term incentive of immediate profits is offset by the long term loss of viability coming from displeased customers.

If the advertisers are the primary victims, and ad networks can sometimes be defrauded too, how do the fraudsters turn a profit? One obvious option for them would become publishers and run their own web sites to attract real users and generate traffic. But this would require quite a bit of work and would hardly be fraudulent... On the other hand, generating fake clicks toward publisher web sites that do not belong to them would only generate revenue for those publishers and not the fraudsters. In order to capture revenues while minimizing web content creation, fraudsters capitalize on SE syndication. Since SE do not have their own content, it is easy for fraudsters to create a web site resembling an SE with minimal web content creation efforts. Using SE syndication from ad networks provides a mechanism for fraudsters operating these SE to sell traffic through these SE. The bots then generate "searches" on these SE, who then incorporate sponsored links that can be sold as CPM or CPC. Because these SE are the entry point for fraudulent traffic, we call them *doorway search engines*. Typically, these SE only exist to lend an air of legitimacy to click fraud and have no real users. We provide a pictorial depiction of this kind of fraud scheme in Figure 2.

III. Collection methodology

In order to study the click-fraud ecosystem, we propose to observe on the activities of botnets by collecting network traces of infected machines over time. These network traces can then be analyzed to reconstruct the redirection chains traversed by these bots in their click-fraud activity. Aggregation of these redirection chains by regrouping sites belonging to the same actors allows us to reconstruct a graph of actors involved, willingly or not, in the click-fraud activities of these botnets.

A. Boaxxe

Boaxxe is the code name for a well-known and documented click-fraud botnet [7]; it also known as *Miuref*. It was first found in the wild in 2012. Boaxxe is a single-purpose botnet doing only click fraud. Unlike other botmasters who recruit machines with a *pay-per-install* model, Boaxxe's botmaster employs a network of *affiliates* who install the Boaxxe bot code on compromised machines they have infected or bought. These affiliates install the code with a hard coded affiliate ID, which is then used by the botmaster to track and remunerate its affiliates for fake click traffic generated from their machines.

To perform click fraud, Boaxxe uses two modes: clickjacking and automated click fraud. In clickjacking (a.k.a. click hijacking), the malware intercepts search requests and clicks made by the real user in order to replace the target of these clicks and requests by advertisement provided by Boaxxe. In automated click-fraud, the malware simply generates traffic in the background, to make it appear as if the user of the infected machine is clicking on ads. For our study, we decided to focus solely on automated click-fraud.

In this mode, Boaxxe launches multiple click-fraud threads. Each of these threads starts by contacting a doorway SE, presumably controlled by the Boaxxe botmaster, such as `asearchit.com`, `tersearch.com` or `fesearch.com`. A screen capture of one of these SE is presented in Figure 3. The reply from the doorway SE is a redirection URL that contains the affiliate ID (in the `subid` variable) and a search keyword (the `q` variable). This keyword is provided by Boaxxe and is passed on to the syndicator ad network, in order to make the redirect look like a legitimate search result. Indeed, it is important to note that the choice keyword can influence the sale price of a click. When the infected computer browses this URL, it enters the advertisement ecosystem and a long chain of redirection through various actors begins. The redirection chain ends on the advertiser's web site, the *landing page*.

We believe that Boaxxe is a good representative of the automated click-fraud monetization scheme. Some investigations have indicated that other click-fraud malware, such as Pigeon, Alureon and Wowlik, also rely on doorway SE [20]. However, the full investigation of the differences in the market surrounding the Boaxxe malware and other automated click-fraud modules is left for future investigations.

Unfortunately, due to the limited availability of Boaxxe samples, we had to rely on a single malware affiliate ID during the course of our study. While we have no reason to believe this introduces a significant bias in our results, this represents a limitation of our study.

As our Boaxxe sample has been unpacked, we were able to run it on a Windows virtual machine, regularly restored to keep it clean. Thus, no additional HTTP traffic was present in the network traces, thus avoiding the need for a step of data pre-processing.

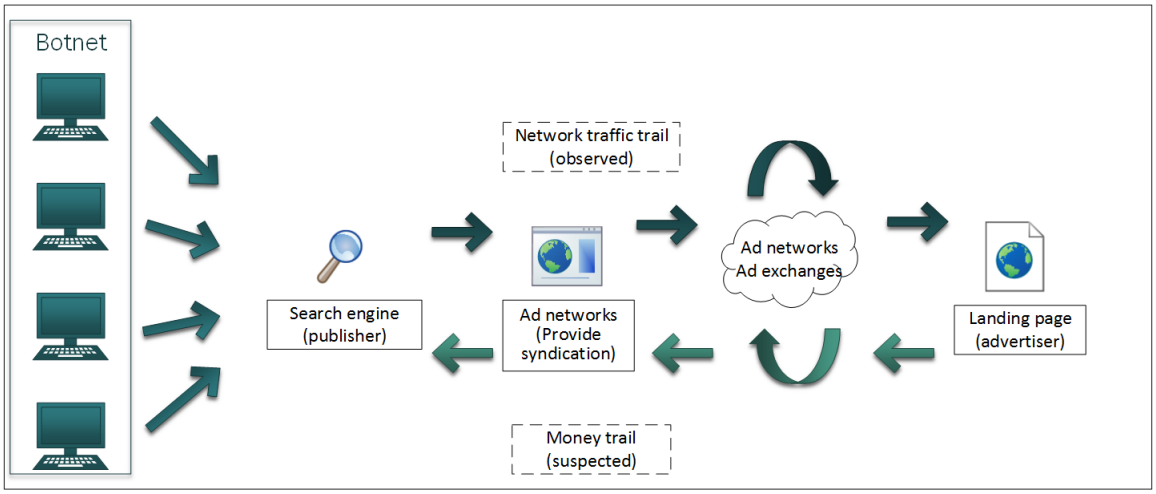

Fig. 2. Traffic and money flows for a click-fraud scheme using a doorway search engine.

Fig. 3. `tersearch.com`

TABLE II
DATA SUMMARY OF BOAXXE LONGITUDINAL STUDY. AS SHOWN IN TABLE I MANY REDIRECTIONS ARE WITHIN THE SAME SITE. THE EXTERNAL REDIRECTION COUNT INCLUDES ONLY THOSE TRANSITING FROM ONE DOMAIN TO ANOTHER.

Total size (PCAP)	3.8 GB
Duration	207 days
Number of chains	1380
Number of external redirections	3218

B. Longitudinal study

Knowing the advertising market is composed of a series of campaigns, each lasting a limited period of time, we chose to make a *longitudinal* study. This means that we collected network traffic traces of a Boaxxe-infected bot frequently (daily) during a long period of time instead of collecting large amounts of traffic over a short period of time. The data collection spanned seven months, from April 2015 to the end of October 2015. We collected ten minutes a day for the first two months and thirty minutes a day subsequently. This way, we could observe the daily click-fraud tasking for the bot. A summary of our dataset is given Table II.

Additionally, we also collected one month of 24/7 data collection in January 2016 in order to measure the potential influence on our results of the relatively short daily collection periods. Furthermore, during this control experiment, we have also regularly changed the geographical location of our exit points to measure the potential influence of location on our results.

The control experiment appears to confirm that the results collected in our longitudinal experiment are valid. The longer daily collection greatly increased the quantity of data collected, but the large majority of that data was redundant in terms of actors identified and their relationships. However, geographical location did seem to have an impact on results. Notably, in some non-English speaking countries, no automated click-fraud activity could be observed. Conversely, other activity not related to click-fraud was observed for United States IP addresses. We did not determine the nature of this traffic, but we suspect it might have been related to search-engine optimization (SEO). Nonetheless, the automated click-fraud activity, when present, was consistent. This would suggest that, while some differences based on geographical location were observed, they do not detract from the generality of our observations related to automated click-fraud for Boaxxe.

C. Chain reconstruction

As seen in Section II-A, we can use the redirection chain to act as a proxy for the value chain of the advertisement ecosystem. Unfortunately, the data collected is raw packet capture files (pcap). It is necessary to extract the HTTP redirection chains to study click fraud. While it is easy to extract individual HTTP requests from a pcap file, it is difficult to link the individual HTTP requests to specific advertisement redirection chains. This is mainly due to the number of redirection chains that occur at the same time by simultaneous threads, and to the variety of redirection types observed in advertisement chains. In particular, a number of redirections are dynamically generated by JavaScript. Moreover, because of the presence of HTTP 300's redirections, it is not possible to blindly trust referer information, as shown in the third column of Table I.

In order to solve these problems, we developed an algorithm to reconstruct the redirection chains. Basically, the algorithm

parses the content of each HTTP packet to retrieve URLs. Then, using information gathered from the URLs, each HTTP request is linked to the HTTP response which triggered the redirection. This generates a tree for each thread, where the root is the initial request to doorway SE. The nodes of this tree are the URLs that were subsequently requested, with each edges represent such a request from the parent URL to the child URL. These requests include the advertisement redirection chain, but also requests to auxiliary resources, e.g. images, CSS, page counters, etc. Once the tree is built, we then extract the advertisement redirection chain. It is simply the path between the root of the tree and the landing page. To find this path, it is necessary to identify which node corresponds to the landing page. Since advertiser landing pages typically provide rich content, they contain many requests to internal and external resources. In contrast, we observed that intermediary nodes (corresponding to ad networks) generally only contain redirection scripts and few or no resources. This allows us to identify landing pages relatively easily by counting the number of children.

Once all the redirection chains are reconstructed from the various trees, we merge the chains into a single graph by regrouping all nodes that share the same domain name or IP address. In that graph, the nodes are the domain names or the raw IP addresses extracted from the redirection chain nodes. The presence of an edge between two nodes means that a redirection between these nodes was found in at least one chain.

D. Node aggregation

This graph cannot be considered an accurate depiction of the business relationships between actors involved in Boaxxe click fraud, because actors typically operate several IP addresses and domain names. In order to make it more useful, we would prefer if the nodes represented actual actors. Thus, we should merge all the nodes belonging to the same organization into a single node. To do so, we developed a methodology based on Whois data, passive DNS data, tracking codes and page similarity.

First, we collected the data from the Whois database on each web site to gather the registrant's name, address, email address and phone number. We also gathered the authoritative name server, when it was not a registrar or hosting service. Two web sites registered to the same company, at the same address and using the same email are likely to belong to the same actor. However, special care must be taken when automating this process because many web sites, including legitimate web sites, use Whois anonymizer services, especially to protect their email address, which can lead to inappropriate merging of nodes.

Second, we used the Virus Total passive DNS service [34] to retrieve the IP addresses resolved by each domain name. Two domain names resolving to the same IP may be an indication that they belong to the same actor. Again, care must be taken because of shared hosting or denial-of-service protection services. Thus, we also verified that the IP address did not belong to a known cloud provider like Amazon EC2 or a DoS protection service such as CloudFlare. In some cases, we also considered the DNS Start of Authority (SOA) record because it contains not only name servers, but also an email address.

Third, we parsed the index file of each web site to retrieve tracking codes, which are account numbers for affiliate programs, a technique developed by Seitz [27]. Because these codes are used to produce sensitive information on page counts (e.g. Google Analytics) or to attribute revenue (e.g. Google Adsense), they are not normally shared across organizations and are unique to each. As such, they are a good indicator that a node belongs to a particular actor. We chose to use the five following affiliate programs: Google Analytics, Google Adsense, Amazon, ClickBank and AddThis.

Finally, in some cases, we noted that several web sites shared the same web page in which only the domain name was changed. By looking at the source code of these web pages, we could confirm that they were indeed identical. We considered only original web pages to avoid default configuration pages. Similarly, we found that several web sites shared the same SSL certificate. While normally web sites with different domain names cannot share a certificate (due to same origin policies implemented in modern browsers), we found that some sites add the same SSL certificate when we crawled them with HTTPS. This is a good indicator that the same default template, including the SSL certificate, was probably used in constructing these sites. Thus, their reuse suggest that they belong to a single actor.

To ensure the quality of this process, we performed a manual check of each merge. In other words, we collected data automatically to support node merging, but we manually confirmed each merge. Overall, this aggregation process allowed us to reduce the number of nodes in the initial graph from 523 to 225 potential actors.

The resulting *actor graph* is the graph where all redirection chain aggregated and all nodes are merged as described above. We postulate that this graph represents an adequate sampling of the overall business relationships between actors in the Boaxxe ecosystem, i.e. *who* is doing business with whom. In addition, we can calculate the weights for each edge based on the number of redirection chains that transited between the corresponding merged nodes. However, due the limitation of our sampling collection methods, we cannot claim that this weighted actor graph gives an accurate description of volume of Boaxxe click fraud, nor on the relative importance of these relationships in terms of fraud revenue and cost. In other words, we cannot quantify *how much* business is being done with whom.

IV. Results

In this section, we present the results of analysis of the actor graph reconstructed as described in Section III-D from the raw data gathered during from the longitudinal study described in Section III-B.

A. Actor graph

Using the chain reconstruction process presented in Section III-C and the node merging procedure described in Section III-D, we reconstructed the actor graph of Boaxxe's automated click-fraud ecosystem. The resulting actor graph is depicted in Figure 4. The visual representation of this graph in Figure 4 is generated by a force-directed method. This means that the algorithm attempts to place nodes on a planar surface in order to minimize "repulsion" between nodes, while modeling the edges as "strings" attaching the nodes. This often results in more strongly connected nodes being placed in the center of the representation, with less connected pushed to the periphery.

As result of the node aggregation procedure, all domain names for the doorway SE were merged into a single node, which we call the *Boaxxe root* or simply root as it has no incoming edges. By analyzing the graph data, we observe that immediate neighborhood of the root relatively small. The components directly linked to the Boaxxe search engines (neighborhood of radius 1) represent only 5.36% of the nodes in the actor graph. This suggests that they may become a good choice for disruption. Graph density is the proportion of edges present in a graph in comparison with a fully connected graph of the same size. In our case, the relatively low graph density of of the radius-1 neighborhood (0.348) suggests that disruption operations are tractable by targeting a limited number of these nodes.

Another observation is that the nodes of the network are regularly reused. Specifically, 58% of the nodes were visited more than once. The average number of times a node was visited is 13 times, with a standard deviation is 40. Prior to merging, our graph was composed of 523 distinct nodes, that were regrouped into 225 nodes. Of those 113 were landing pages and 11 were in the immediate neighborhood of the Boaxxe root. It is important to note that all actors in the immediate neighborhood were identified after 73 days of data collection, suggesting than our sampling probably provides sufficient coverage of the key intermediate actors in the Boaxxe click-fraud ecosystem. There is much more volatility in actors corresponding to landing pages, however, which is natural given the fact the much larger population of advertisers with respect of ad network intermediaries and the volatile nature of their advertising demand (i.e. campaign-driven advertisement).

B. Actors

Because Boaxxe's click-fraud scheme is organized around a syndicated doorway SE, it is useful to delve more deeply in the ad networks that are offering it syndication. After all, it seems fairly evident that Boaxxe is involved in malicious activity. Even if the search engines used by Boaxxe regularly change domains, they always use the same HTML page. Furthermore, any deep inspection of the traffic would reveal that the traffic is generated by bots. For instance, the search results obtained from the doorway SE have nothing to do with the search requests. As such, it is reasonable to suspect that the ad networks directly linked to the doorway search engines are buying botnet traffic knowingly.

We can look at some examples of ad networks directly connected to the Boaxxe SE, i.e. at a distance of 1, to understand this category of actors. One of these ad networks is called *Nextadnet*. Based on the Whois information of web sites owned by Nextadnet, we were able to determine that this company is based in Cyprus. However, when browsing their web site, the only methods offered to contact them are via Skype, e-mail, ICQ and Jabber, which is unusual even for an Internet company. Moreover, someone claiming to be a representative of that company posted an offer to buy traffic on *blackhatworld* [4], a forum of questionable reputation, known for providing information and tools for Black Hat SEO techniques. It would be surprising if that was down with the intention of acquiring good-quality traffic for their advertisers.

Another example is the *superior-movies.com* web site, which was regularly used as the first redirection after the doorway SE during the first four months of our study. Its homepage consists of several movie trailers, none of which refer to recent movies. However, if the web site is browsed with a particular set of URL parameters, it will redirect the user to an IP address of the media company *Daoclick* instead of landing on the homepage of `superior-movies.com`.

These two examples paint the portrait of typical fly-by-night advertisement companies that knowingly deal with illicit or unethical actors.

In comparison, the ad networks with a distance of two from Boaxxe are a mix of well-known media companies, like `advertise.com` or `ad.com`, domain parking services, such as *Parking Crew* or *Go Daddy parking*, and ad networks deliberately and publicly offering low quality traffic, like `popcash.net`. For the most part, these are well known companies with a presence in the legitimate market.

Looking at the landing pages, we observe well-known web sites, like Amazon, Bing or the Huffington Post, but also shady web sites like `fasttcash.biz` that proposes get rich quick schemes, or `valortechhelp.com`, a web page containing nothing but ads. Interestingly, we also found within the landing pages in the first months of our study the same Bonnier group mentioned in the Bloomberg article cited in Section II. We can also confirm from our dataset that the traffic received by Bonnier during that period came from `advertise.com`, as was also described in the Bloomberg article. Overall, we found that 12% of the landing pages in our dataset were part of the Alexa top 10,000. Another 29% are out of the top 10,000, but are still within the Alexa top 1,000,000. The fact that 41% of all observed Boaxxe traffic ends up in Alexa top 1,000,000 web sites, most of which are presumably legitimate web sites, strongly underlines the fact that Boaxxe traffic has no difficulty entering the legitimate advertisement market.

V. Social Network Analysis

While some insight may be gained from looking at the actor graph alone, if our goal is the disruption of the click-fraud ecosystem, we need to perform more in-depth analysis.

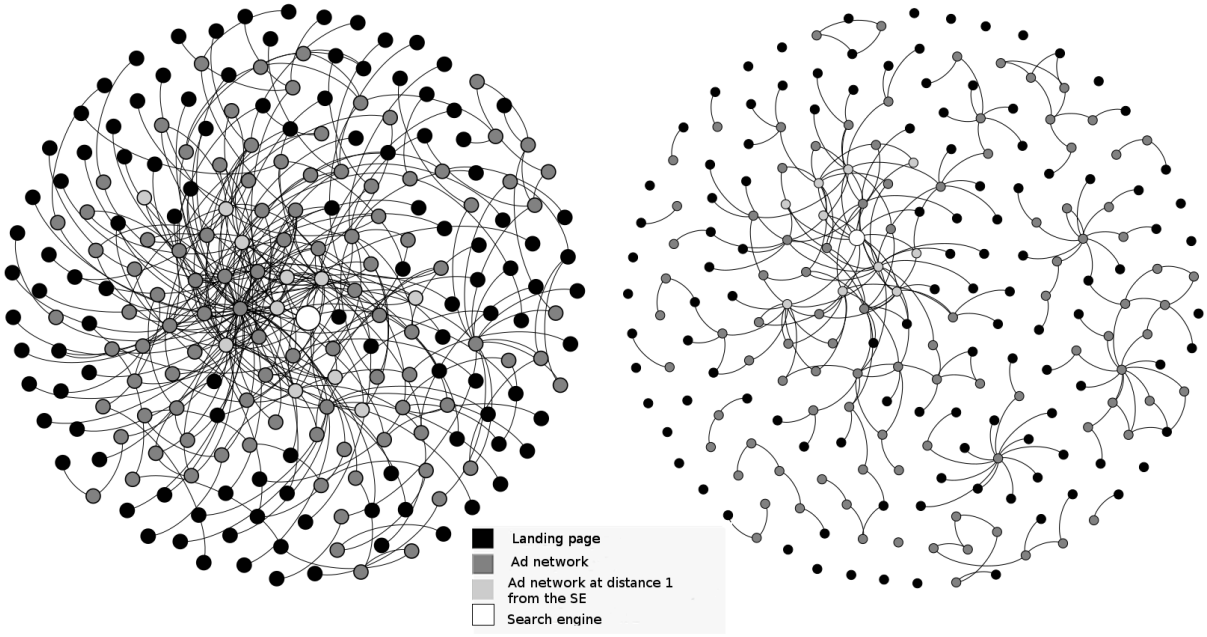

Fig. 4. Comparison of the Fruchterman-Reignold force-directed representation of the non-disrupted (left) and disrupted (right) click fraud ecosystems.

In this section, we discuss a selection of disruption methods that could be employed to disrupt the ecosystem, and present on the theoretical performance of one of them, the *keyplayer* technique.

A. Disruption

Our ultimate goal is to disrupt the click-fraud ecosystem. In other words, we want to reduce the ability of the operators of the Boaxxe malware to inject traffic into the legitimate advertising market and money from it. As the Boaxxe ecosystem is represented by our actor graph, it is possible to evaluate the effect of eventual disruption operations on the actor graph. Multiple methods can be used to attack the graph. Initial research about botnet disruption by Davis *et al.* [12] evaluated random, tree-like and global strategies. Unfortunately, we cannot directly apply these techniques as they rely on inherent properties of peer-to-peer command and control graphs. Similarly, other findings involving the use of Sybil attacks [13] are inappropriate for disruption of the click-fraud ecosystem as it would require acquisition, at great cost, of massive amounts of traffic.

Instead of using techniques developed in the context of disrupting malware infrastructure, we can instead leverage the work of criminologists to disrupt networks of criminals. Notably, it is possible to rely on Social Network Analysis (SNA), a technique shown to be effective when dealing with cyber criminal networks by Décary-Hétu and Dupont [14]. The goal of this technique is to find ways to disrupt such the criminal ecosystem by analyzing the network of social and business relationships between criminal actors. In this kind of analysis, the most important actors in the network, or key players, are identified. This is critical for precisely targeting disruption efforts. This was emphasized by Clayton *et al.* [8] who argue that poorly targeted disruption operations allow for quick recovery by cyber criminals.

Many graph metrics exists in SNA to help identify the central players in a network [16]. The most common metrics used are *centrality* and *betweenness*. Centrality metrics are a measure of the direct influence of actors through their direct ties to alters. Betweenness metrics focus more on the indirect ties between actors, and measures the extent to which network paths pass through the actors. An actor who is often used to relay information between pairs of actors will have a high degree of betweenness, although this may not represent his real power in the underlying criminal network. Neither of these commonly used metrics are appropriate in the case of click fraud.

First, fraudulent traffic could still be channeled through other actors and redirection paths even if the highest centrality ad network is "removed". As long as alternate redirection paths exist, fraudulent traffic can be sold. The ecosystem may offer less flexibility to the fraudsters, but it will remain able to accomplish its purpose: monetize fraudulent traffic. Second, targeting ad networks with high betweenness scores may lead to inefficient removal of actor. Betweenness metrics capture the criticality of edges in terms of how graph distances are affected when they are removed. In the case of click fraud, an increase of distance due to the disruption of high betweenness actors would might force the fraudulent traffic to transit through more intermediaries to reach the same landing pages. This would hardly constitute an important disruption to monetizing operation in most cases.

To circumvent the limitations of traditional metrics, Bor-

gatti [6] defined the Key Player Problem (KPP). In its Negative formulation (KPP-Neg), the objective is to optimize disruption by measuring the effect on the cohesiveness of the successive removal of key players from the graph. Basically, it aims at calculating the influence of removing a given set of actors on the normalized fragmentation of the graph, i.e. the number and size of connected components of the actor graph. A normalized fragmentation value of 1 means that all the nodes are disconnected. On the other hand, a fully-connected network would have a fragmentation value of 0. In this context, the removal of a node represents an actor no longer participating in the activity modeled by the graph. A more in-depth discussion of techniques to convince actors to cease their activities will be given in Section VI.

In order to compute the KPP-Neg problem, we used `Keyplayer2`, a program developed by Borgatti [5]. It implements a KPP-Neg solver, with a choice of three heuristics. In order to obtain results in a timely fashion, we chose to use the Greedy heuristic. We opted for 5,000 iterations to improve the results.

B. Keyplayer Analysis Results

When looking at the results, we can see that removing the first three nodes has a large influence on the cohesion of the graph. At 4 nodes removed, the normalized fragmentation is above 80%, a high level of fragmentation. Moreover, it seems that removing more than 3-5 nodes has diminishing returns as the fragmentation plateaus. The high fragmentation obtained by removing a small number of actors is encouraging as this implies disruption could be relatively easily achieved.

The first three selected nodes for removal are *AdKernel*, *Deximedia* and *Vertamedia*. Once these three nodes are removed, the network becomes much more fragmented. The resulting graph is shown in Figure 4. By looking at the graph, it is clear that the removal of these three nodes does not isolate the Boaxxe root from its immediate neighborhood of radius 1. However, it does somewhat isolate that neighborhood from many legitimate ad networks and landing pages. Consequently, it becomes far more complex for Boaxxe to monetize their traffic.

However, we also need to make sure that it would not be possible for cyber criminals to quickly and efficiently replace these nodes. Therefore, we need to further analyze the nature of these three nodes removed by the algorithm.

AdKernel: This is an ad network Solution-as-a-Service (SaaS) provider. They provide all the infrastructure required to run an ad network. They are not necessary directly involved in click-fraud, but their service is used by many suspicious ad networks. Table III summarizes the ad networks found in our dataset that use AdKernel services, with customers found in the Alexa Top 100,000 ranking in bold. Most of the ad networks present in the Alexa Top 100,000 ranking should be reputable companies. However, the highest Alexa-ranked ad network, *Vertoz* in this list, is known also for being involved in malvertising [23]. Moreover, when looking at the Virus Total Passive DNS database, we noticed that a number of additional ad networks with suspicious practices not present in our dataset also use AdKernel services.

Deximedia: There is limited information available on the web about this US media company except for a job offer in New York City and a discussion topic on the *blackhatworld* forum [3]. This anecdotal evidence would seem to imply that Deximedia is not an ad network with a strong policy against click fraud. As such, it is difficult to know how much effort would be required by the underground to replace this node if it were removed from their ecosystem.

Vertamedia: The headquarters of this media company are located in New York City, but the two co-founders appear to work from the Ukraine. According to its web site, this company has 20 employees and seems to participate in different digital media events. Despite the fact that they are directly connected to the Boaxxe doorway SE, this suggests that they have a well-established position in the legitimate advertising market. However, one of their domains, `c.feed-xml.com`, was embedded in the strings of several malware samples, including the well-known click-fraud malware Poweliks [35]. Finally, this domain was also associated with Bedep, another malware with a click-fraud payload [17].

The capacity to affect actors that are directly involved with Boaxxe or otherwise involved in illicit activities might be limited. Furthermore, affecting targeting those actors might be inefficient as they can easily be replaced by fraudsters. For example, if we consider removing more than 2 nodes, we encounter some of the ad networks (Vertamedia) that are directly linked to the Boaxxe root. These are usually not attractive targets for disruption because of their fly-by-night nature.

For this reason, we considered an alternate application of the keyplayer technique where certain nodes were "protected" from the KPP-neg solver; we call these the *untouchable actors*. We ran the KPP-Neg solve again with all actors in the immediate neighborhood of the Boaxxe root labeled as untouchable. This enabled us to identify intermediary targets that are at a distance greater than one from Boaxxe.

Once this is done, it is interesting to note that some well-known ad networks appear in the list of nodes whose removal produces a high fragmentation delta, notably `advertise.com`, *eZanga* and *BlueLink Marketing*. `advertise.com` was already singled out for accepting click-fraud traffic by Bloomberg in the previously cited article II-B. In the same manner, eZanga and BlueLink Marketing were already suspected to buy botnet traffic in 2013, as detailed in an Adweek interview of Web fraud expert Ben Edelman [1]. In summary, these ad networks appear to have recurring issues with botnet traffic. They would thus seem to be attractive choices for disruption, as their positions in the legitimate ad market would make them ideal targets for pressure from their legitimate clients.

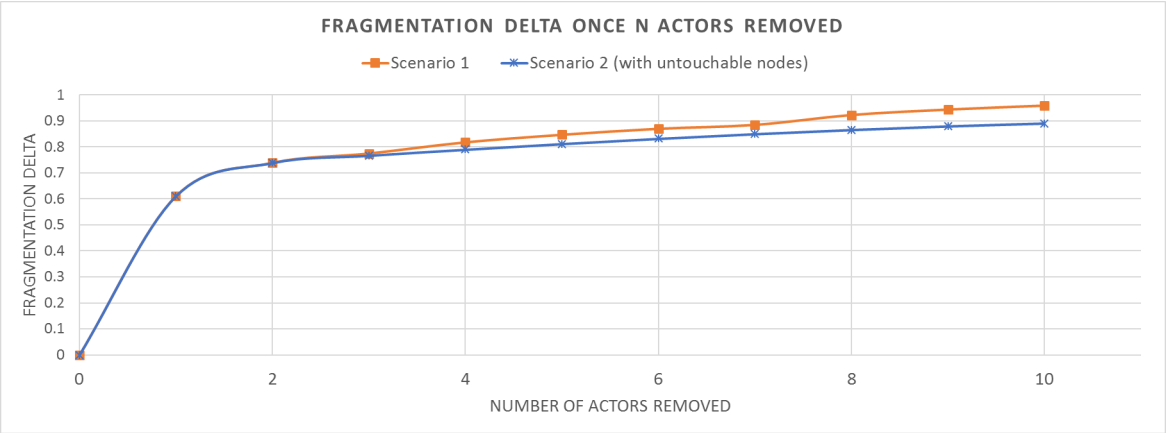

Fig. 5. Impact of removal of actors on the fragmentation of the actor graph as measured by the difference in normalized (fragmentation delta, y-axis) once a given number of actors (x-axis) are removed. In Scenario 1 any node can be removed, while in Scenario 2 nodes in the immediate neighborhood of the Boaxxe doorway SE are *untouchable* and cannot be removed.

TABLE III
LIST OF ADKERNEL CUSTOMERS

bluemediappc.com	terappc.com	dsmedianet.com	eliteppc.net	olmeramarketing.com
adsparkmedia.net	vokut.com	infinitywidget.com	finditquick.com	anytheengmedia.com
searcharbor.com	castramedia.com	marsfeeds.com	primusad.com	maxppc.com
vertoz.com	mindad-xml.com	mediacpc.com	ybrant-search.com	readywind.com
dogtownads.com	seodollars.com	**vespymedia.com**	madeofmedia.com	trafficaim.com
ctrtraffic.com	visitorsblitz.com	zipzipsearch.com	cpc-ads.com	globalsearchmedia.com
resultscpv.com	adconfide.com	xmladsystem.com	infinity-info.com	cubxml.com

VI. DISCUSSION

A. Interpretation of results

At first glance, the most efficient way to stop fraudulent activity from a botnet would be to perform a take-down operation. However, take downs are resource consuming and offer limited guarantees for long-term effectiveness. As an example, the ZeroAccess botnet was taken down in 2013 and resurrected at the beginning of 2015 [30]. As long as monetization schemes allow botnet operators to generate profits, they will continue to reinvest a good portion of them to rebuild any botnets taken down. In that light, disrupting the monetization scheme appears to be the best way to achieve long term results.

In the Boaxxe case, it is evident that the doorway search engines are the most critical nodes because they represent the root of the redirection tree. However, as seen in our dataset, the Boaxxe operator(s) already routinely changes the search engine domains. If the search engines are simply taken down, it would require very little effort to setup substitutes. That is not the case for the other economic actors involved in the Boaxxe click-fraud ecosystem. Even the fly-by-night advertisement companies that provide syndication for the Boaxxe search engines require incorporations and a veneer of legitimacy. Without those characteristics, it would not be possible to inject the click-fraud traffic in the legitimate market, where the victims reside.

In the previous section, we presented a technique to disrupt the Boaxxe's click-fraud ecosystem. Our results show that by removing only three carefully selected nodes we could impede the majority of click-fraud traffic to reach its victim. Indeed, we found that more than 50% of the landing pages would be disconnected from the doorway SE. This small number of targeted actors would seem to imply that the resources required for a successful disruption operation could be much smaller than the resources required to perform a traditional botnet take-down. If more resources were available, a crippling disruption of the ecosystem could be achieved with less than ten targeted actors. Furthermore, even if the actors that are closer to the Boaxxe root cannot be targeted (i.e. are "untouchable"), for example because they reside in unfriendly jurisdictions, it is still possible to obtain a significant disruption by targeting accessible actors. After all, as seen in our results, the difference in fragmentation delta between the scenarios with and without untouchable nodes is relatively small.

B. Targeting actors in practice

The keyplayer analysis technique allows us to identify the best candidates for disruption in the graph. However, it provides no guidance in how to "remove" the corresponding actors from the graph. Several options can be considered. First, legal action could be taken. While click-fraud is may not be explicitly illegal in the criminal sense, there are no doubts that it represents a breach of contract in many cases.

Second, some of the ad networks that we identified in our dataset have well-known customers that could apply pressure on their providers. These ad networks enable cybercrime, whether willingly, by virtue of their negligence or by a lack of

ethical guidelines. They contribute to the success of criminals and undermine the digital ecosystem. For instance, as shown in our data, the Huffington Post, a well-known publisher, receives traffic from Deximedia. If customers such as the Huffington Post demanded stricter action against click-fraud by their traffic providers such as Deximedia, these ad networks would be required to comply, or else lose the business from these customers. To do so, advertisers should also change their practices to assess the quality of the traffic they receive. In particular, we advocate for the use of better metrics to measure the Return-on-Investment (ROI) on Internet publicity. The prevalent use of metrics such as volume of incoming traffic, page counts, etc. is an indirect cause of this publicity fraud phenomenon. Furthermore, we also need to remember that ad networks frequently sell traffic to each other. As such, other ad networks could threaten to ban bad apples from their ad exchanges or standing purchasing agreements in a similar manner as discussed above for advertisers.

Third, we showed that the targeting of AdKernel, for example, an ad network SaaS provider, caused the most impact on the click-fraud ecosystem. This is not surprising as it drastically decreases the barriers to entry in the advertising market. With this kind of SaaS service, anyone can launch an ad network, or even relaunch it when its reputation becomes too poor. In this light, SaaS providers could be made to accept more accountability for the activities of their customers, and could offer help in removing any known bad actors that are abusing their services.

Finally, the advertising industry launched in 2014 the Trustworthy Accountability Group [33]. It aims to regulate the advertising market by giving a certification to companies that can be trusted. Moreover, they developed *Payment ID*, a system in which each click or impression is given a unique identifier. Thus, when an advertiser detects invalid traffic, he can follow the supply chain and blacklist the fraudulent traffic providers at the origin of the redirect chain.

VII. Related work

While no other research group has specifically tackled the problem of disrupting the click-fraud ecosystem, a number of researchers have provided insight on the world of click fraud.

One of the first analysis of a click-fraud malware binary was that of Clickbot.A in 2007 [9]. The authors detail the low-noise techniques used by the malware operator to perform click-fraud and present an estimation of the cost of the fraud for advertisers. While this malware did not cause any damage to the infected computer or its owner, the authors claim that ad networks, anti-virus companies, advertisers and publishers should work together to disrupt such activities. The rationale is that these activities generate a large amount of money for criminals and create incentives for them to cause harm to users. Later, Miler *et al.* [24] examined two different click-fraud malware, 7cy and Fiesta, in order to compare them with Clickbot.A. They found new techniques employed by this malware to mimic the behavior of a human browsing web sites in order to evade fraud detection. In 2014, Pearce *et al.* [26] made a detailed analysis of the ZeroAccess click-fraud malware and its monetization strategy. However, the paper was limited to describing the ecosystem rather than to find ways to disrupt it. Thomas *et al.* focused on studying ad injections, a form of advertisement fraud involving extensions that modify the web page DOM [32].

Alrwais *et al.* [2] studied the effect of the FBI's Operation Ghost Click. This was a large take down of an ad-fraud botnet that was using rogue DNS server to hijack valid ads and replace them by ads supplied by the malware, a form a clickjacking. However, this operation did not disrupt the advertising ecosystem related to the malware. Chances are that parts of this ecosystem have been be reused by other click-fraud malware since.

Other studies have focused on the ad ecosystem itself. Stone-Gross *et al.* [31] explored an ad exchange system to understand how these systems can be abused by criminals to generate profit. Zhang *et al.* [38] bought traffic from different traffic providers for their own web site and evaluated, for each provider, the quality of the traffic. They found that the traffic coming from bulk providers was of poor quality in comparison to Google Adwords. Snyder *et al.* [28] studied affiliate marketing fraud. Dave *et al.* [10][11] focused on how to detect fraudulent clicks by using appropriate metrics. Recently, Javed *et al.* [19] showed the existence of traffic exchange services that provide an alternate way of generating fraudulent clicks to automated click bots.

VIII. Conclusion

In this paper, we described the click-fraud ecosystem of Boaxxe/Miuref, a well-known click-fraud botnet. We collected click-fraud network traces from self-infected Boaxxe bots in a 7-month longitudinal study. By reconstructing the redirection chains of the automated click-fraud activity, it was possible to adequately sample the actor graph of the ecosystem.

We then applied Social Network Analysis to find the key players of the fraud ecosystem. We found that by removing a very limited number of actors the monetizing capacity of the botnet could be seriously disrupted. Of these actors, one of the most interesting is assuredly AdKernel, a Solution-as-a-Service provider for ad networks. This is not is not surprising as it enables companies to enter in the advertising market by reducing barriers to entry. This illustrates the importance of preventing the use of these services by criminals.

Finally, as click fraud and other types of ad-based monetizing schemes become an increasingly important source of revenue for criminals, we argue that ecosystem disruption techniques based on information acquired from the analysis of redirection chains should be more widely used. While botnet take downs can achieve short term success, they are less efficient in the long term.

An important limitation of our work is the fact that we cannot guarantee that the redirection chain reconstructed from client network traces captures *all* of the business relationships involved in Internet advertisement, and hence on click fraud; it is only a subgraph of real graph of actors involved in

Boaxxe click fraud. While the collected traffic represents only a very small fraction of the overall botnet traffic, we have discussed in Section IV the reconstructed graph probably provides adequate coverage of the key players in the *visible* graph. However, we cannot exclude the possibility that certain click purchase transactions could happen without leaving a trace in the redirection chain, e.g. in the case of sales by commission between trusted parties. The only way to identify potential key players not visible in the actor graph would by cross-referencing our data with that of actual ad networks, for example through industry-wide data sharing initiatives, such as those mentioned in Section VII.

Future work should widen our study. As seen before, an ad network directly linked to the Boaxxe search engines was also seen in Bedep traffic, another click-fraud malware. This suggests that it could be worthwhile to collect network traces from several click-fraud botnets and apply the same key player method and compare the results, in order to see whether the same disruption targets apply to several click-fraud botnets. Similarly, the method could be applied to other ad-based monetization schemes such as black search engine optimization and adware. This more global study might enable us to disrupt more generally the ad-based fraud ecosystem that is a threat to the web economy.

IX. Acknowledgments

This work was funded by the Natural Science and Engineering Research Council of Canada (NSERC), the *Fond Québécois de Recherche Nature et Technologie* (FQRNT), PROMPT and ESET Canada through the Collaborative Research and Development (CRD) program. The authors would like to thank Nedra Hamouda at École Polytechnique de Montréal for her background research on Internet publicity at-large and Internet publicity fraud in particular, that provided valuable quantitative and qualitative information on the phenomenon in support of this work. In addition, the authors would like to thank Pierre-Marc Bureau, from Google, for valuable comments and feedback.

References

[1] AdWeek. The six companies fueling an online ad crisis, 2013. http://www.adweek.com/news/advertising-branding/six-companies-fueling-online-ad-crisis-150160.

[2] 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*, pages 21–30. ACM, 2012.

[3] Blackhatworld.com. What is deximedia.com?, 2014. http://www.blackhatworld.com/blackhat-seo/facebook/659769-what-deximedia-com.html.

[4] Blackhatworld.com, 2015. http://www.blackhatworld.com/blackhat-seo/other-ppc-networks/759368-we-looking-new-traffic-sources.html.

[5] S. Borgatti. Keyplayer program. http://www.analytictech.com/keyplayer/keyplayer.htm.

[6] S. P. Borgatti. Identifying sets of key players in a social network. *Computational and Mathematical Organization Theory*, 12(1):21–34, Apr. 2006.

[7] J. Calvet. Boaxxe adware: 'a good ad sells the product without drawing attention to itself' pt 1, 2014. http://www.welivesecurity.com/2014/01/14/boaxxe-adware-a-good-ad-sells-the-product-without-drawing-attention-to-itself-pt-1/.

[8] R. Clayton, T. Moore, and N. Christin. Concentrating Correctly on Cybercrime Concentration. In *Proceedings of the Fourteenth Workshop on the Economics of Information Security (WEIS)*, Delft, Netherland, 2015.

[9] N. Daswani and M. Stoppelman. The Anatomy of Clickbot.A. In *Proceedings of the First Conference on First Workshop on Hot Topics in Understanding Botnets*, HotBots'07, pages 11–11, Berkeley, CA, USA, 2007. USENIX Association.

[10] V. Dave, S. Guha, and Y. Zhang. Measuring and fingerprinting click-spam in ad networks. *ACM SIGCOMM Computer Communication Review*, 42(4):175–186, 2012.

[11] V. Dave, S. Guha, and Y. Zhang. Viceroi: Catching click-spam in search ad networks. In *Proceedings of the 2013 ACM SIGSAC Conference on Computer & Communications Security*, CCS '13, pages 765–776, New York, NY, USA, 2013. ACM.

[12] C. Davis, S. Neville, J. Fernandez, J.-M. Robert, and J. Mchugh. Structured peer-to-peer overlay networks: Ideal botnets command and control infrastructures? *Computer Security-ESORICS 2008*, pages 461–480, 2008.

[13] C. R. Davis, J. M. Fernandez, S. Neville, and J. McHugh. Sybil attacks as a mitigation strategy against the storm botnet. In *Malicious and Unwanted Software, 2008. MALWARE 2008. 3rd International Conference on*, pages 32–40. IEEE, 2008.

[14] D. Décary-Hétu and B. Dupont. The social network of hackers. *Global Crime*, 13(3):160–175, Aug. 2012.

[15] B. Elgin, M. Riley, D. Kocieniewski, and J. Brustein. The fake traffic schemes that are rotting the internet, 2015. http://www.bloomberg.com/features/2015-click-fraud/.

[16] S. F. Everton. *Disrupting dark networks*, volume 34. Cambridge University Press, 2012.

[17] S. Frankoff. Sentrant — Bedep Ad-Fraud Botnet Analysis - Exposing the Mechanics Behind 153.6M Defrauded Ad Impressions A Day, 2015. https://sentrant.com/2015/05/20/bedep-ad-fraud-botnet-analysis-exposing-the-mechanics-behind-153-6m-defrauded-ad-impressions-a-day/.

[18] Interactive Advertising Bureau and Price, Waterhouse, Coopers. 2015 full year - digital advertising revenue report, April 2016.

[19] M. Javed, C. Herley, M. Peinado, and V. Paxson. Measurement and analysis of traffic exchange services. In *Proceedings of the 2015 ACM Conference on Internet Measurement Conference*, IMC '15, pages 1–12, New York, NY, USA, 2015. ACM.

[20] P. Kalnay and J. Horejsi. Notes on click-fraud: American story. In *Virus bulletin conference*, pages 118–129, 2014.

[21] T. Krazit. Yahoo settles pay-per-click fraud suit, 2009. http://www.cnet.com/news/yahoo-settles-pay-per-click-fraud-suit/.

[22] B. Krebs. Chronopay's scareware diaries, 2011. http://krebsonsecurity.com/2011/03/chronopays-scareware-diaries/.

[23] Malekalmorte. directrev malvertising lead to Zbot | malekal's site, Jan. 2014. http://www.malekal.com/directrev-malvertising-lead-to-zbot/.

[24] 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 (DIMVA)*, pages 164–183. Springer, 2011.

[25] B. Mungamuru, S. Weis, and H. Garcia-Molina. Should ad networks bother fighting click fraud?(yes, they should.). In *Stanford InfoLab, Technical Report*. Stanford, 2008.

[26] 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 *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*, pages 141–152. ACM, 2014.

[27] J. Seitz. Automatically Discover Website Connections Through Tracking Codes | Automating OSINT Blog, Aug. 2015.

[28] P. Snyder and C. Kanich. No please, after you: Detecting fraud in affiliate marketing networks. In *Proceedings of the Workshop on the Economics of Information Security (WEIS)*, 2015.

[29] Statista. Digital advertising spending worldwide from 2014 to 2016, 2016.

[30] M. Stockley. Zeroaccess click fraud botnet coughs back to life, 2015. https://nakedsecurity.sophos.com/2015/01/31/zeroaccess-click-fraud-botnet-coughs-back-to-life/.

[31] 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*, pages 279–294. ACM, 2011.

[32] K. Thomas, E. Bursztein, C. Grier, G. Ho, N. Jagpal, A. Kapravelos, D. McCoy, A. Nappa, V. Paxson, P. Pearce, and others. Ad injection at scale: Assessing deceptive advertisement modifications. *Security and Privacy. IEEE*, 2015.

[33] Trustworthy Accountability Group. Home, 2016. https://www.tagtoday.net/.

[34] Virustotal, 2015. https://www.virustotal.com.

[35] Virustotal. c.feed-xml.com domain information, 2016. https://www.virustotal.com/en/domain/c.feed-xml.com/information/.

[36] White Ops and Association of National Advertisers. The bot baseline: Fraud in digital advertising. Dec. 2014.

[37] Wordstream. Average cost per click around the world, July 2015.

[38] Q. Zhang, T. Ristenpart, S. Savage, and G. M. Voelker. Got traffic? an evaluation of click traffic providers. In *Proceedings of the 2011 Joint WICOW/AIRWeb Workshop on Web Quality*, pages 19–26. ACM, 2011.