TraffickStop: Detecting and Measuring Illicit Traffic Monetization Through Large-scale DNS Analysis

Baojun Liu[1,2], Zhou Li[3], Peiyuan Zong[4], Chaoyi Lu[1,2], Haixin Duan[1,2,5], Ying Liu[1,2], Sumayah Alrwais[6], Xiaofeng Wang[7], Shuang Hao[8], Yaoqi Jia[9], Yiming Zhang[1,2], Kai Chen[4] and Zaifeng Zhang [10]

[1] Tsinghua University, [2] Beijing National Research Center for Information Science and Technology,
[3] University of California, Irvine, [4] Institute of Information Engineering, Chinese Academy of Sciences,
[5] 360 Enterprise Security Group, [6] King Saud University, [7] Indiana University Bloomington,
[8] University of Texas at Dallas, [9] Zilliqa Research, [10] 360 Netlab

***Abstract*—Illicit traffic monetization is a type of Internet fraud that hijacks users' web requests and reroutes them to a traffic network (e.g., advertising network), in order to unethically gain monetary rewards. Despite its popularity among Internet fraudsters, our understanding of the problem is still limited. Since the behavior is highly dynamic (can happen at any place including client-side, transport-layer and server-side) and selective (could target a regional network), prior approaches like active probing can only reveal a small piece of the entire ecosystem. So far, questions including how this fraud works at a global scale and what fraudsters' preferred methods are, still remain unanswered.**

To fill the missing pieces, we developed `TraffickStop`, the first system that can detect this fraud *passively*. Our key contribution is a novel algorithm that works on large-scale DNS logs and efficiently discovers abnormal domain correlations. `TraffickStop` enables the first landscape study of this fraud, and we have some interesting findings. By analyzing over 231 billion DNS logs of two weeks, we discovered 1,457 fraud sites. Regarding its scale, the fraud sites receive more than 53 billion DNS requests within one year, and a company could lose up to 53K dollars per day due to fraud traffic. We also discovered two new strategies that are leveraged by fraudsters to evade inspection. Our work provides new insights into illicit traffic monetization, raises its public awareness, and contributes to a better understanding and ultimate elimination of this threat.

I. Introduction

Web traffic (or visits) generated by Internet users is the "lifeline" of the Internet economy. Oftentimes, the revenue of an Internet company directly depends on the amount of traffic it receives. To help websites get traffic more easily, *traffic networks* have emerged. For instance, advertising network tunnels traffic from publisher sites to advertiser sites, and reward publishers who forward "organic" (i.e., user-generated) traffic.

Nevertheless, the monetary reward from traffic networks also attracts fraudsters who earn illicit profit from manipulating the web traffic. Their common approach is to tamper with users' web requests and reroute them to a fraudulent site, till getting to a contracted traffic network. Such an underground business model (i.e., *illicit traffic monetization*) has become so profitable that a broad spectrum of illicit activities have been discovered on client-side (e.g., malware infection), transport-layer (e.g., ISP hijacking), and server-side (e.g., fake search ads [32], [65]). Counting the fraud against the pay-per-view (PPV) network only, the annual loss inflicted on advertisers who pay for invalid traffic could be more than 180 million US dollars [72].

Prior work. Despite of its significant impact, our current understanding of this problem is still limited, due to the lack of effective methods to detect such fraud at a large scale. Existing approaches are mainly "active" to discover such malicious activities, including deploying honey ads [30], embedding inspection JavaScript [68], [74], and probing the network [29], [49]. However, those approaches either require a deep involvement of publisher websites [74], or work on only one type of fraud like ISP content injection [61], which only constitutes a small part of the entire fraud landscape.

Our methodology. To acquire a broader view of affiliate frauds and an in-depth understanding of their security implications, we developed a novel *passive* methodology that discovers frauds through analyzing users' DNS traffic logged by the DNS recursive servers around the world. Through "connecting the dots" in traffic data, we are able to detect more illicit activities and gain a more comprehensive view of their strategies. Similar passive analyses have shown successes in other research areas like botnet detection [24], [25], [26], [66]. However, they cannot be directly applied here, since the infrastructure of affiliate fraud is different, and prominent features like DGA, domain/IP fluxing and short domain-lifetime are not effective any more in our setting.

We look at this problem from a unique angle. More specifically, our key observation is that a fraudulent site redirects their visitors *only* to the site running affiliate program (called program site) in a *short* period of time. As a result, DNS requests between the fraud site and the program site exhibit *strong but anomalous* correlations. This inspires us to develop a module called *Association Finder*, to analyze DNS traffic and discover suspicious domain relations (e.g., `hao.125y.com` → `hao.360.cn`). To make the analysis efficient, we optimized a classic algorithm called `FP-Growth` on the MapReduce programming model and incorporate the decay factor for association analysis for the first time.

Due to the limited information contained in the DNS traffic, the domains discovered by Association Finder are not always

[*] Haixin Duan and Ying Liu are corresponding authors. This work was done during Baojun Liu's research internship at Netlab of 360.

fraudulent. As such, we built another module called *Content Analyzer* to filter the detection results. Our key observation is that unlike legitimate publisher sites hosting rich and meaningful content, fraud sites typically do not care about content quality. Instead, all they want is just to *automatically redirect* visitors to program sites. As such, we crawled URLs from the suspicious domains to identify their redirection behaviors. Content-based clustering was applied separately for websites which use suspicious templates to commit ad fraud.

Our detection system, `TraffickStop`, consists of the two modules and we used it to analyze a passive DNS dataset with **231 billion** requests from **13 million** IPs. In the end, our system reported **1,457** fraud sites conducting traffic fraud against three types of traffic networks (eCommerce, navigation, and advertising). Our evaluation of `TraffickStop` shows that the accuracy could reach 72.7% (89.4% for eCommerce, 67.5% for navigation and 74.8% for advertising, respectively). In addition, looking into the detected sites, our system is able to uncover the adversaries sitting on *entirely different stacks* (*client-side*, *transport-layer* and *server-side*). Such detection coverage cannot be achieved by any existing detection systems based on active probing.

Our findings. On top of the detected fraud sites, we performed a comprehensive measurement study, shedding light on the strategies and ecosystem of the affiliate fraud. Below are a few highlights of our findings:

1) Scale of the problem. From the statistical information obtained from a Passive DNS service [19], we analyzed the impact of traffic fraud on Internet users (especially those in China). In particular, we found that fraud sites could receive a huge number of visits, at more than **53 billion times** in just one year. Among the fraud sites, 232 (15.9%) sites receive more than 10M DNS requests, and huo99.com, which is the doorway site for a homepage hijacking campaign, accounts for **19 billion** DNS requests from 05/2017 to 04/2018.
2) Types of adversaries. We investigated each type of adversaries to understand their operational models. On the server side, we found that *Search Ad Impersonation* is extensively conducted by fraudsters, who aggressively buy Baidu's search keywords and display misleading search ads to attract visitors. On the transport layer, we observed that most requests to a fraudulent FQDN tend to come from a small region, which indicates that the regional ISP could be involved in the illicit activities. On the client side, we discovered 8,555 kinds of PUPs and malware which conduct traffic fraud, and many of them are masked under *gaming softwares*.
3) Economy impacts. Using the pricing policies published by traffic networks or those commonly agreed, we built a model to estimate the losses inflicted on some networks. Our study shows that a network can easily lose from **53K** (Taobao eCommerce ad) to **1K** (360 Navigation) US dollars per day.
4) Evasion strategies. We discovered two new strategies called *domain renting* and *ad reselling*, which are leveraged by fraudsters to evade inspection of traffic networks.

Paper Organization. The remainder of this paper is organized as follows. We provide background in Section II and collect groundtruth dataset for empirical study in Section III. We elaborate system design and evaluation in Section IV and Section V. Section VI presents our measurement findings. And Section VII presents discussion about our detection system. We summarize previous related work in Section VIII. And finally, Section IX concludes the paper.

II. Background

In this section, we first present an overview of how web traffic is monetized under different business models. Then, we elaborate on how different adversaries could defraud traffic networks through traffic manipulation.

A. Traffic Monetization

Gaining large volume of visits from web users is a critical goal for website owners. Web visits are usually *referred* by other well-known sites, like search engines. The connection between site owners and traffic referrers are usually established by *traffic networks*, who purchase traffic from referrers and then tunnel it to customer sites for monetary reward. The traffic referrers are called *affiliates* and they share the reward with the partnered traffic network. Below, we elaborate three types of traffic networks and their business models[1].

eCommerce Network. Affiliates in this case are obliged to promote commodities of the partnered merchants. Their approaches are flexible, like writing good reviews on their websites. Commission fee is paid to the affiliates when sales are made *after* their sites are visited by buyers. While large eCommerce sites, like Amazon, run their dedicated affiliate networks, many online retailers share eCommerce networks, like Rakuten LinkShare [67], to reach out to the buyers.

When an affiliate registers itself in a network, an affiliate ID will be assigned, together with URL links pointing to the merchandises. These links, when embedded in the affiliate's website and clicked, will redirect visitors to the merchant's sites. The affiliate ID is embedded in visitors' requests, e.g., through setting HTTP cookies [27], and logged by the merchant to identify the affiliate.

Advertising Network. Online advertisements (or ads) is the most popular monetization method. A publisher is allowed to show multiple ads at the same time on its web page, which increases the chance of a successful referral. How ads are selected and where they are displayed are handled by the advertising network. Currently, the main business models include 1) *search advertising*, which displays ads along with search results, and 2) *contextual advertising*, which embeds ads in publishers' pages. How much an affiliate can earn depends on the volume of views, clicks or accredited user actions (e.g., making a purchase).

As the online ads ecosystem has evolved to contain diverse publishers, advertisers and networks, *ad syndication* is invented to connect these entities for resource sharing. It works in a hierarchical structure: e.g., the first-tier ad network (e.g., doubleclick) can ask a second-tier one (e.g., clicksor) to serve an ad, and then shares the revenue. At user's end, ad syndication is completed through *automated* redirection (e.g., 302 HTTP redirection).

[1] Other terms have been used to refer to the traffic networks covered in this work. For example, eCommerce network is also called affiliate marketing [27]. To emphasize the business type of customers and avoid confusion, we choose different terms.

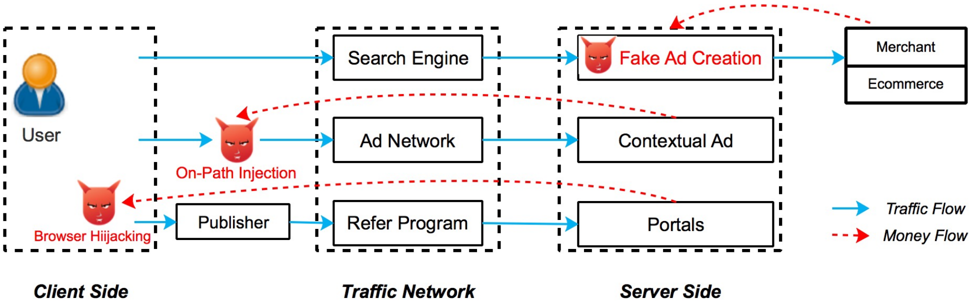


Fig. 1: Traffic flow and money flow in three adversary models.

Navigation Network. Web portals, which bring websites together to help users navigate information, also look for incoming traffic. These web portals have their own navigation networks and partner with many affiliates, especially in China. As an example, hao123.com, a portal site run by Baidu, partners with hundreds of affiliates and serve requests coming from hundreds of millions of web users [9].

The portfolio of affiliates is usually diverse. Taking hao123.com as an example again, its affiliates include browser vendors that set their default homepages to hao123.com, site owners that refer visitors, and PC-end software that recommends hao123.com as browser homepage or promotes the hao123 desktop assistant [10]. In addition, it also runs navigation network on mobile platform [11]. Affiliates are rewarded based on the number of visitors they bring in per day, which are identified by their IP addresses. While an advertising affiliate is required to show ads to visitors before they land on customer sites, such step can be skipped by navigation affiliates. And ss a result, visitors are automatically sent to web portals.

B. Adversary Models

A traffic network should filter out traffic of bad quality before it reaches the customers, and penalize cheating behaviors. To this end, many rules have been made and enforced by traffic networks [1], [2], [3], [18]. Examples are: 1) traffic must be generated by human rather than a bot or scripts; 2) traffic must not be manipulated without user's consent; 3) publisher's content should be honest and meaningful (e.g., pages of the publisher should not be empty). In addition, eCommerce and navigation networks forbid using iframe for redirection [71].

However, catching cheating affiliates in the network is always challenging. Traffic networks have little control over visitors' devices, network connections and affiliates' servers. As a result, *illicit traffic monetization*, the underground business that sells bad-quality traffic and earns profit unethically, is thriving. In this work, we studied three types of adversaries who commit the fraud. In principal, all activities involve a *fraudulent site* (or *FS*) that redirects traffic to a website belonging to a traffic network (called *program site*, or *PS*), and violate at least one policy. Below we elaborate the common strategy used by each adversary (also illustrated in Figure 1).

Server-side: Search Ad Impersonation. In this setting, the adversary registers an advertiser account of a search engine and publishes fake search ads. In order to trap more visits, the search ads usually impersonate well-known brands, which are shown in the prominent places of search result pages. In May 2014, over 300 fake ads were found on Bing, Yahoo and Google by The Search Monitor, and many leading companies were defrauded [32], [65]. We provide an example in Appendix A.

Transport-layer: ISP Injection. Before reaching its destination, traffic from users goes through many middle-boxes where packets can be tampered by adversaries. In this work, we focus on ISP injection, where an ISP attempts to load extra ads on users' devices by injecting code into web responses. ISP injection can be mitigated by forcing HTTPS, yet the adoption rate of HTTPS is still low in some regions, like China [39].

Client-side: Browser Hijacking. Installing PUP (potentially unwanted software) or malware on users' machines elevates attackers' power to manipulate web traffic and makes mitigation much more difficult. Traffic networks and server-side parties know little about client-side behaviors, and transport-layer protection like HTTPS is out of scope. In this work, we studied how adversaries defraud traffic networks by browser hijacking. For instance, visits to one web portal from a browser could be rerouted to another web portal to steal traffic from users [4].

III. Understanding Illicit Traffic Monetization

A. Data Collection for Empirical Study

We bootstrap our study by collecting web pages and URLs of FS in the wild. As far as we know, there is no public dataset available. Therefore, we collected a list of FS manually and visited them to create the ground-truth dataset.

For Search Ad Impersonation, we created a list of search keywords of popular commercial products (e.g. "washing machine", "jackets" and "tennis racket") and queried them on Baidu Search for four months. We looked for the sponsored ads impersonating well-known brands. In the end, 57 FS were found to steal traffic through fake ads. For Browser Hijacking, we searched for relevant posts in several forums [7], [8], [16] and identified 50 FS. We manually browse various technique forums, where browser hijacking is discussed and malicious URLs are shared in the posts. We believe the forums provide us with good sources to collect ground truth manually. The list of search keywords will be provided for other researchers to extend our work.

On the other hand, collecting ground-truth data for ISP injection is more difficult: adversaries usually perform selective injection [21] based on where visitors come from. To obtain vantage points with good network coverage, we "out-sourced" this task to Internet users in China. Specifically, we embedded our code into an *Adobe Flash advertisement* and submitted it to an advertising platform. On the client side, user fetched the advertisement content from our web server. And our script will send back the received content. The returned content were compared to the untampered versions to discover discrepancy. If differences are identified, the returned page is considered to be tampered and the HTML content is considered for further analysis. As a result, 44 FS were collected. Here, we take utmost care to protect users from side-effects that may be caused by our experiment. We pay for the advertising platform and totally abide by their terms of service. According to the agreement with the advertising platform, every steps in this process are allowed.

In total, we collected 151 FS regarding the three adversaries and call this dataset D_{ES}.

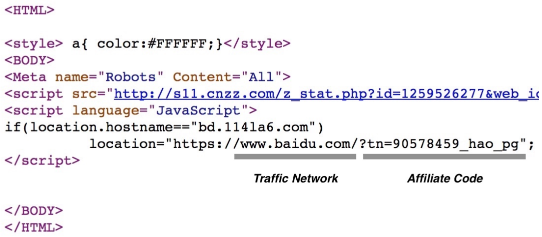
Ethics. The experiment running on users' devices could raise ethical concerns, and we address them by carefully designing the measurement code and routines. Firstly, data sent back to our server only contains contents injected by ISP, without any users' private information. Secondly, our ads are only for experiment purposes without promoting any merchants. Different from using ads for cryptocurrency mining, we do not gain any profit from the activities. Since the only task of our flash ad is to send injected contents to our server, the computational overhead on users' end is small. Thirdly, all collected data is sent back encrypted and securely stored on servers of our lab. Only personnel working on this project have access to the server/data. To notice, previous works [28], [56], [75] also runs advertisement measurement code on users' devices, and our treatment of privacy aligns with these works.

B. Observations

Traditional detection mechanisms of malicious websites (e.g., web content analysis) are effective when they behave maliciously (e.g., downloading malware). However, in our case those approaches are not effective, as an FS only forces redirection which also happens on legitimate websites. As such, new features are needed to distinguish FS in the context of traffic exchange, and they should also be robust against content- and URL-level evasion. Below, we elaborate two key features we discover through our empirical analysis of collected FS.

Strong domain correlation. We observed that PS is the *only* redirection destination for the majority of FS, and the redirection always happens *automatically* and *immediately* (125 out of 151 FS in D_{ES} [2]). This setting requires no action from users and maximizes the amount of referred visitors. It also results in *strong correlation* between FS and PS. By contrast, redirection targets of legitimate websites are usually more diverse. For instance, a visitor of a news site might visit other sections, see ads from other hosts or go to a completely different site subsequently.

On the other hand, strong domain correlation also exists when a PS redirects to other sites of the same owner, for load balancing purposes as an example. However, those sites tend to share the same ownership information and can be filtered out with the help of WHOIS data.



```
<HTML>

<style> a{ color:#FFFFFF;}</style>
<BODY>
<Meta name="Robots" Content="All">
<script src="http://s11.cnzz.com/z_stat.php?id=1259526277&web_i
<script language="JavaScript">
if(location.hostname=="bd.114la6.com")
        location="https://www.baidu.com/?tn=90578459_hao_pg";
</script>


</BODY>
</HTML>
```

Fig. 2: Webpage of an FS.

Meaningless content. Different from legitimate sites whose owners are striving to provide useful content and attract more visitors, the main functionality of an FS is only redirection. As a supporting evidence, web pages of 143 FS in D_{ES} only contain HTML tag or JavaScript code which conducts redirection, without meaningful content on the same page. In fact, content in FS bears little value, as the duration of user stay is very short due to automatic redirection. According to policies of traffic networks, serving meaningless content is not permitted (see Section II). Another interesting discovery is that many fraudulent pages share a same HTML structure. We speculate they are constructed using popular web templates or toolkits.

Here we illustrate one example FS, bd.114la6.com, whose web page is shown in Figure 2. We can observe from the page that baidu.com is the targeted PS and *90578459_hao_pg* is the affiliate code. The embedded JavaScript code will be executed once the web page is loaded, redirecting to Baidu. Obviously, the page contains no meaningful content.

IV. System Design and Implementation

Leveraging the two insights we gain from examining D_{ES}, we built TraffickStop, a system that is able to detect FS by analyzing large-scale network data. In this section, we first give an overview of its architecture, followed by elaborating the implementation of each component.

A. Design Overview

Ultimately, TraffickStop aims to accurately and timely identify FS through passive analysis. It works on a large-scale dataset of network logs to achieve comprehensive coverage (finding adversaries at server-side, client-side and transport-layer). Active approaches including network probing [29], [49] and code embedding [68], [74] either target a single adversary (e.g., transparent-layer) or require substantial changes on traffic networks, which cannot achieve the same goal as ours.

In particular, we choose DNS logs collected from resolvers in the wild as the data to be processed by TraffickStop. Security companies including FarSight Security [38] and 360 Security [19] have been collecting passive DNS data for a long time, and providing open access for research purposes. Compared to DNS data, HTTP logs provide better visibility but they are not used in our study, as the data is much more sensitive and we have not found a large-scale public dataset.

Figure 3 abstracts the workflow of our system. As the first step, TraffickStop extracts necessary fields from DNS logs and filters out irrelevant records. The pre-processed data goes to a module named *Association Finder*, which attempts to

[2] The remaining ones fail to redirect to a valid page of PS because we were not able to construct the correct URL parameters.

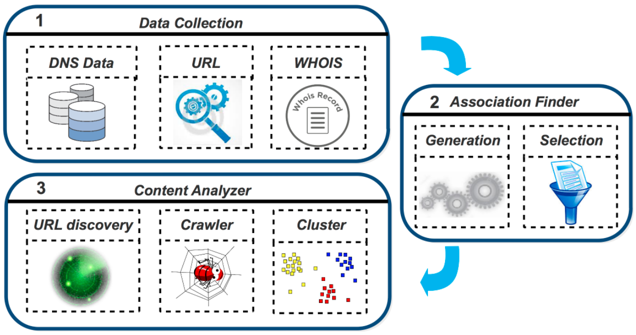


Fig. 3: The workflow of `TraffickStop`.

find pairs of domains exhibiting strong correlation by *association analysis*. Subsequently, pairs containing monitored PS are forwarded to a module named *Content Analyzer*. This module crawls the URLs hosted under suspicious sites (those other than PS) to discover *automated redirection* to PS. Suspicious sites are directly labeled as FS if the target PS runs eCommerce or navigation programs, as automated redirection is forbidden in the policies (described in II-B) [1], [3], [18]. Otherwise, `TraffickStop` runs cluster analysis to determine if the suspicious site is engaged in advertisement fraud.

B. Association Finder

The goal of this component is to find domain pairs $\{X, Y\}$ with strong correlation. In particular, such a pair has to meet three criteria: 1) X and Y appear together with high frequency; 2) When X is observed, Y can be observed with high probability; 3) The visit interval between X and Y is small.

The first two criteria match the metrics *support* and *confidence* used by *association analysis* [44] perfectly, so we adopted association analysis in `TraffickStop`. How *support* and *confidence* are computed are shown below (N is the number of all records).

$$Support, s(X \rightarrow Y) = \frac{\sigma(X \cup Y)}{N} \tag{1}$$

$$Confidence, c(X \rightarrow Y) = \frac{\sigma(X \cup Y)}{\sigma(X)} \tag{2}$$

To find frequent associations, two algorithms `Apriori` and `FP-Growth` are widely adopted [45]. `Apriori` is easy to implement but incurs high overhead when processing large dataset. Therefore, we chose `FP-Growth`, which first scans the dataset twice to construct a compact structure called `FP-tree`, and then finds associations with high *support* and *confidence* (over thresholds called $minsup$ and $minconf$). To model our third criterion, we introduced an extra *decay* factor when computing *confidence*.

The original `FP-Growth` algorithm is able to capture associations with multiple entities (e.g., triplets). Since we are only interested in pairs, we optimized the structure of `FP-tree` and removed long associations to reduce storage overhead. In practice, `TraffickStop` is implemented on top of MapReduce to compute in parallel. The details are elaborated as follows.

Data pre-processing. To begin with, `TraffickStop` extracts three necessary fields from each entry of DNS dataset: pseudonymized client IP address, timestamp and queried domain name. Table I presents a sample of the pre-processed data.

TABLE I: A sample of DNS Log data.

Timestamp	Anonymized IP	Queried Domain Name
1488383983	c747....8e05	www.16163.com
1488383984	cf2e....34a2	ex-std-node625.prod.rhcloud.com
1488383984	cf2e....34a2	av-caoba.rhcloud.com
1488383983	3452....8e21	nga.cn
1488383983	61a5....bc8d	eastmoney.xdwscache.ourwebcdn.com

In the following process, IPs associated with large number of distinct FQDNs are removed. This is because such IP may be NAT IP or proxy IP which represent many users. Under this circumstance, sessions of multiple users are tangled, and the visiting sequence of domains does not reflect the authentic visiting sequence of an individual user, dampening the effectiveness of *Association Finder*.

Data sorting. To find out the domain visiting sequence of a client IP, we need to sort our dataset by its timestamps. In practice, a parallelized sorting process is implemented on top of MapReduce. Specifically, data is first sliced into buckets labeled by IP (in *Map* function) and then sorted in parallel on multiple machines (in *Reduce* function). The sorted data is further sliced by hour (called IP-hour bucket) to enable fast data loading. In addition, to reduce computation overhead, we convert each requested FQDN to its effective second-level domain (SLD) (e.g., `www.google.com` is converted to `google.com`) [3], if they are not under known CDN SLDs (identified by matching a public list [13]). We single out the CDN case because adversaries can host FS on CDN domains to reduce the degree of correlation, although such case has not been found in our empirical study.

In the paper, we use MapReduce as a representative case, only to demonstrate that our algorithm is carefully optimized for parallel processing, and can be deployed on distributed platforms. Our system can also be deployed on other big data frameworks.

Pair discovery. Again, we resorted to MapReduce to identify domain pairs with strong correlation. The pseudo code is listed in Algorithm 1. For *Map* function, each IP-hour bucket is loaded and we use a sliding window to confine the interval between two domain visits. Assuming the window has a fixed number of domains (which is used to estimate the interval), we first selected the domain close to the middle point of the window as the key (also called *association source*), and then used all other domains within the window as values (also called *association destination*), to construct key-value pairs. Particularly, we allow association destination to be ahead of the association source, in order to capture FS under the circumstance of DNS prefetching.

The original version of `FP-Growth` does not model the *distance* between a source and its destination. In our case, the smaller the distance, the higher chance two domains are visited during an automated redirection. Here, we leveraged the *decay* function as a metric of the distance, since it has shown

[3] We used a public suffix list [20] for the transformation, so domains under known public suffixes, e.g., `www.example.co.uk`, will not be mistakenly transformed into `co.uk`.

Algorithm 1 Pair discovery based on FP-Growth.

```
Input: Sorted DNS data
Output: Rule, confidence, support
 1: function MERGE($Group\_source$)
 2:     for $uniq\_dest \in destination\_set$ do
 3:         $confidence \leftarrow$ SUM_VALUE(uniq_dest)$/source.support$
 4:         $Rule[uniq\_dest] \leftarrow uniq\_dest.support, confidence$
 5:     return $Rule$
 6:
 7: Procedure: Map
 8: for $DNS\_Sequence \in DNS\_database$ do
 9:     while $index < DNS\_Sequece.length$ do
10:         $source \leftarrow DNS\_Sequece[index]$
11:         $session \leftarrow DNS\_Sequece[index{-}window, index{+}window]$
12:         for $destination \in session$ do
13:             $value \leftarrow$ DECAY(source.location, destination.location)
14:             Out: $source$, $destination$, $value$
15:         $index++$
16:
17: Procedure: Reduce
18: $Group\_source \leftarrow$ GROUPBY($source$)
19: $Rule \leftarrow$ MERGE($Group\_source$)
20: $Rule\_group \leftarrow$ FILTER_RULE($Rule, minsup, minconf$)
21: for $rule \in Rule\_group$ do
22:     Out: $source\_domain$, $destination\_domain$, $confidence$, $support$
```

successes in previous studies [59]. In the function described by the formula below, $X.ind$ and $Y.ind$ are the indices of domains within the window. λ is the decay factor, with the value setting to 0.10 based on empirical analysis. The *decay* value is computed within *Map* and is outputted together with source and destination.

$$Decay(X.ind, Y.ind) = 2^{-\lambda|X.ind - Y.ind|} \tag{3}$$

The key-value pairs are later distributed into buckets labeled by association source and processed by *Reduce* function, which counts the frequency of pairs and sources to compute support and confidence. Here, we changed the confidence function (equation 2) to use the *decay* value instead of frequency (line 3 of Algorithm 1). Therefore, domains that are always visited within short intervals will have high confidence value. In the end, pairs surpass the thresholds are sent to the next step.

Domain ownership analysis. As discussed in Section III-B, a pair of domains might show strong correlation if they are registered by the same entity. Such domain pairs are legitimate, so we checked the similarity of their WHOIS information and filter them out before content analysis.

On the other hand, an advanced adversary might copy all WHOIS fields from a PS to her FS to bypass our check. This is feasible for most WHOIS fields except *registrant email address*, which should be receiving verification messages according to ICANN policy [46]. However, we find not all registrars in China follow this policy, where no email verification is performed to update domain WHOIS. Therefore, we also considered the *name server* field, since adversaries have no incentive to set name server of an FS to those serving PS, in which case FS will not receive any domain request.

Target selection. TraffickStop intends to find fraudulent activities against known traffic networks. Other kinds of malicious activities exhibiting similar characteristics (e.g., redirection between compromised and exploit sites) are not considered. As such, we filtered out a domain pair if its association destination does not match a monitored PS. The remaining pairs are considered suspicious.

C. Content Analyzer

This component analyzes the domains other than PS in the suspicious domain pairs detected by Association Finder. In essence, it looks into each domain's web content and checks whether it launches automated redirection or serves meaningless content. Below we elaborate the implementation details.

URL discovery. A legitimate site usually serves a homepage under its SLD. However, this observation does not hold for FS, where the fraudulent page is often hosted on a subdomain of the SLD. In addition, filename has to be specified in the URL request oftentimes. As such, an auxiliary data source is needed to uncover the URLs published by an FS. In this work, we leveraged a URL dataset provided by a security company (elaborated in Section V-A) and selected the top 10 most popular URLs under each SLD.

Dynamic crawling. To analyze contents of each URL, we need to retrieve its webpage and check whether it conducts automatic redirection without user's consent. While static crawlers, e.g., Scrapy [69], can be used for the first task, it is not suitable for checking redirections, as they can be triggered by obfuscated JavaScript code. Therefore, we choose to build a dynamic crawler based on an open-source crawling framework named OpenWPM [37], which runs on top of Firefox and logs all browser behaviors and HTTP data. All pages visited by OpenWPM and the targets of automatic redirection are stored in a Sqlite database.

If the redirection target matches a PS, the source domain becomes an FS candidate. If the PS belongs to eCommerce or navigation networks, the source domain is directly marked as a true FS, because automated redirection is explicitly forbidden by the policies [1], [3], [18]. For advertising networks, especially contextual ads, we further clustered them by content and perform manual analysis, as automated redirection is legitimate.

Content-based clustering. Web template is frequently utilized to set up FS, which results in many FS sharing the same page structure. By contrast, legitimate pages of different sites are more likely to look different [35]. Therefore, we are motivated to cluster all crawled files based on their structure, and consider large clusters with many web pages as fraudulent.

In practice, we first extracted the tag names from all HTML elements and put them into a sequence, e.g., $\langle iframe, script, ...\rangle$. Then, we parsed all the JavaScript code within the script tag to AST, extract all function and variable names, and also put them into a sequence, e.g., $\langle Fun_{example}, Var_a, ...\rangle$. The first step is skipped if the crawled file is a JavaScript file. Our clustering method is similar to previous works [52], [55] except for one difference: two files fall into the same cluster only when both HTML and JavaScript sequences *exactly* match. Our method is more efficient than the heavy-weight algorithms, like hierarchical clustering, and we still find many clusters. Finally, we manually sampled pages from each cluster and check their content is meaningless.

To notice, we do not perform any supervised or rule-based detection since they are prone to be evaded. For instance, although we find that the number of words is usually small (less than 100) within an FS, we do not transform this observation into a detection rule, as it can be evaded by text spinning [78].

V. Evaluation

In this section, we first describe the datasets used in this study and how the parameters are selected. Then, we elaborate the evaluation result after running `TraffickStop`.

A. Datasets

We give an overview of each dataset as follows (also summarized in Table II).

Finer-grained DNS dataset. We obtained the access to a passive DNS dataset maintained by Qihoo 360 under its DNS Pai Project [33]. This project has been continuously collecting billions of DNS logs daily since 2014, from a large array of DNS resolvers (mostly located in China). The data is open to researchers and all client IP address are anonymized. In this study, we performed analysis on *two weeks* (October 1 ~ October 14, 2017) of its DNS data, containing **231.7 billion** requests from **13.32 million** client IPs. To determine the parameters of Associate Finder, we obtained DNS logs from two universities in China, each with 400 million records approximately.

To measure the scale of traffic fraud (in Section VI), we also obtained one-year aggregated statistics of DNS logs (count of lookups, first and last timestamps of a domain) from May 2017 to Apr. 2018. The dataset contains query logs of around 214K suspicious domain names yielded by Association Finder (1.9GB data).

Previous studies have used passive DNS data to understand and measure malicious behaviors. For instance, 1.45 billion DNS requests are examined by Paxson *et al.* to study surreptitious communication over DNS [62], and 36 billion DNS lookups are examined by Grier *et al.* to assess the damage of exploit kit [41]. The data used by our study (231.7 billion DNS requests) is more extensive comparing to those works.

PS list. We created the list of PS by manually checking Alexa top 500 websites of China and looking for web pages contents describing traffic networks. As a result, 40 PS were selected, as shown in Table III. We assumed a PS is associated with a single traffic network, which holds for all PS in our list.

URL dataset. We obtained URLs under each suspicious SLD from a security company, whose anti-virus software has been installed by millions of users. The software periodically uploads a small subset of URLs observed from user-end to a cloud-based sandbox for in-depth analysis. All URL query parameters are removed by the software to address privacy concerns. As far as we know, the selected URLs are not necessarily malicious and whether they are conducting traffic fraud is not among the uploading criteria. Leveraging this URL dataset, we found 2,118,237 URLs under the suspicious SLDs (188,967 in total) and their subdomains.

Previous researches have used URLs from the user-end for measurement. For instance, WINE platform run by Symantec [73] offers URLs collected from user machines, and the data has been used to measure droppers [50]. We took the similar measure, like data sanitization, to avoid violating users' privacy. While such measure prevents us from detecting FS that only redirects to PS when the request format matches the rule of FS, we still discover thousands of them.

We discuss how the parameters are selected in Appendix B. The parameters include client IP threshold, sliding window size, *minsup* and *minconf*. Parameters including window size and IP threshold are selected from empirical studies, and we carefully adjust the parameters base on our ground truth analysis.

B. Evaluation Result

After processing 231.7 billion DNS requests, our Association Finder outputted 550,090 domain pairs with high correlation. There were 214,015 SLDs labeled as association source. Leveraging domain ownership analysis, 25,048 SLDs were removed, leaving 188,957 SLDs treated as suspicious. After that, we obtained 2,118,237 associated URLs from the URL dataset. However, only 251,878 URLs of them were still alive in May 2018. All web pages of the URLs were crawled by Content Analyzer, and we found that 6,984 pages redirect to PS automatically. Among them, 369 and 1,429 URLs redirected to PS of eCommerce and Navigation networks, so they were directly labeled as fraud. The remaining 5,186 URLs were examined by content-based clustering, to check whether they were used for ad fraud. A cluster is more suspicious if it contains more items (which use similar web templates), so we selected clusters containing at least 4 pages and flagged 32 clusters (499 URLs) as fraud.

The detected 2,465 URLs need to be validated to confirm fraud behaviors. While online scanners, like VirusTotal, can validate malicious URLs related to drive-by-download and phishing attacks, they are not effective for traffic fraud. Therefore, for URL validation, we used three rules following the policies established by traffic networks (described in Section II-A). For each URL, we check 1) whether its content is illegal (e.g., pornographic or gambling pages) or unreadable; 2) whether the page is forcing redirection and 3) whether the URL contains an affiliate ID. If condition 1&3 (for advertising networks) or 2&3 (for eCommerce and navigation networks) are satisfied, we confirm the URL as fraud.

In the end, we find that 1,792 URLs under 1,457 SLDs are fraudulent (72.7% of detected URLs). Among them, 330 URLs (295 SLDs), 963 URLs (812 SLDs) and 499 URLs (350 SLDs) are classified under eCommerce, navigation and advertising fraud (see details in Table IV). The detection accuracy of each category is 89.4%, 67.5% and 74.8% respectively. Compared to previous works [61], [71] based on active probing, our system is able to detect all three types of fraudulent behaviors and achieves better detection coverage. Collecting affiliate code pattern and pre-filtering the URL list can improve detection accuracy, and we plan to test these methods in the near future.

Effectiveness of Association Finder. Since it may take our Content Analyzer several seconds to one minute to complete the analysis of each URL, Association Finder should be able

TABLE II: Data sources. FQDN/day and SLD/sec are measured per client and the median values are listed.

Type	Source		Time Span	Size(~)	Clients	FQDN/day	SLD/sec	Usage
DNS	DNS Pai Traffic		2 weeks	**231.7B requests**	13.32M	/	/	Association Finder
	DNS Log	Unv. T	1 day	372.9M requests	110K	3370.67	1.44272	Pre-measurement for parameters
		Unv. D	5 days	396.4M requests	83K	947.90	1.32457	
	Aggregated DNS		1 year	1.9 GB	-	-	-	Scale and loss analysis
URL	A security company		10 days	2,118,237 URL	-	-	-	Content Analyzer
Blacklist	A browser company		1 day	500 URL	-	-	-	Coverage analysis

TABLE III: PS selected from Alexa CN Top 500.

Types	Count	Exemplary SLDs
Search	6	baidu.com, 360.cn, sogou.com
Navigation	3	hao123.com, qq.com, duba.com
Shopping	14	taobao.com, tmall.com, jd.com
Ads	4	baidustatic.com, jd.com, umeng.com
Jobs	5	zhaopin.com, liepin.com, 51job.com
MISC	13	58.com, dianping.com, meituan.com
Total	40[a]	

[a] An SLD of PS can run multiple affiliate programs so the total count is less than the sum (column 2). The FQDN of a PS only represents one program.

to identify potential FS with higher probability than randomly selecting domains. To verify whether our Association Finder meets such requirement, we randomly sampled 200K domain names from zone files of three TLDs provided by Verisign [76] (for `com` and `net`) and PIR [64] (for `org`), and crawled their associated URLs. 116,041 domains responded to our web requests, with only 14 of them (0.007%) redirecting to PS in our list (eCommerce Network and Navigation Network). In addition, only 2 of them (0.001%) are likely FS (containing affiliate code in the redirection URLs). Both ratios are much lower than the detection results of our Association Finder (3% redirecting to PS and 0.7% are likely FS). As such, we conclude our Association Finder is effective in finding potential FS.

Clustering result. We performed clustering on the detected ad-fraud URLs to reduce the workload of manual inspection. In this process, 28 HTML clusters and 19 script clusters are found. The maximum size of each HTML and script cluster are 56 and 24. Among them, 15 HTML clusters (199 URLs) and 18 script clusters (300 URLs) are true positives as they contain illegal content.

After manually inspecting the remaining clusters, we found the web pages contain many advertisements (e.g., parking pages), show illegal content (pornographic or malware), conduct CPC fraud (the link is not clicked by our crawler but by the embedded script) or only load ads.

Coverage. To assess the detection coverage, we tested `TraffickStop` on a labeled dataset provided by a security company, with 500 URLs under the browser hijacking category. We manually examined the URLs and find 111 URLs (85 SLDs) active during the evaluation period, which are all contacted by PUP. Association Finder identified 58 SLDs (covering 73 URLs) and Content Analyzer reported 48 SLDs (covering 57 URLs) as fraud, resulting in 56.4% recall (48 out of 85). After analyzing the false negatives, we find that 27 URLs and 10 SLDs are missed by Association Finder and Content Analyzer because 1) we do not have URLs associated with those SLDs and 2) our crawler does not carry the desired user-agent string.

Performance. We tested Association Finder on a Hadoop cluster with 100 servers (*only* 1G memory allocated for each server). It took 49 hours to process the entire DNS dataset (over 231 billion requests for the 14-day window). On average, it only took 3.5 hours to process one day's DNS data. For Content Analyzer, we deployed OpenWPM on 10 virtual machines (each with 4G memory) and made them visit all 251,878 suspicious URLs. It took 9 days to finish the task. Although the performance is acceptable for our research, we are planing to improve its performance by pre-filtering URLs, e.g., by considering domain reputation.

VI. Measurement

Using the results of system evaluation, we performed a measurement study to shed light on the ecosystem of illicit traffic monetization. In particular, we assessed the overall scale of traffic fraud and describe our observations on different types of adversaries. Finally, we assessed how much an affiliate program could lose under traffic fraud.

Here we highlight some of our discoveries. In terms of scale, traffic fraud can affect a great number of Internet users, as **53 billion** DNS requests of detected FS are captured within one year. Many fake search ads are placed by fraudsters to hijack search traffic, and they are very aggressive in buying search keywords (more than **two thousand** keywords associated with one FS). PUP and malware are also popular vectors: we find **8,555** kinds that automatically visit FS on user's behalf, and many belong to **gaming softwares**. Traffic fraud can cause significant loss to traffic networks: one network could spend **thousands of dollars** daily in rewarding fraud affiliates.

In addition, our study reveals two new strategies, *domain renting* and *ad reselling*, which are used to bypass the regulations of traffic networks. Details are elaborated in VI-F and VI-G. Also, we present a case study showing how different fraud approaches can work together.

A. Fraud Scale

The first question we want to answer is how this fraud impacts the Internet users, in terms of traffic scale. Described in Section V, 1,457 active FS are detected by our system. Though not exhaustive, we believe this dataset provides a good viewpoint into the landscape.

Looking into the Passive DNS dataset which provides one-year aggregated statistics (described in Section V-A, from 05/2017 to 04/2018), we find the 1,457 FS receive up to **53**

TABLE IV: Result summary of Content Analyzer.

	URL Count	eCommerce Networks				Navigation Networks				Advertising Networks			
		URL	Confirmed	SLD	Accuracy	URL	Confirmed	SLD	Accuracy	URL	Confirmed	SLD	Accuracy
HTML	222,607	369	330	295	89.4%	1,427	963	812	67.5%	4,729	199 (362) [a]	139	55.0%
Script Files	29,271	0	0	0	-	2	0	0	-	457	300 (305)	211	98.4%
Total	**251,878**	369	330	**295**	**89.4%**	1,429	963	**812**	**67.5%**	5,186	499	**350**	**74.8%**

[a] 199 (362) means that we select the clusters with at least 4 items (362 URL) and check them manually. 199 are labeled as fraud.

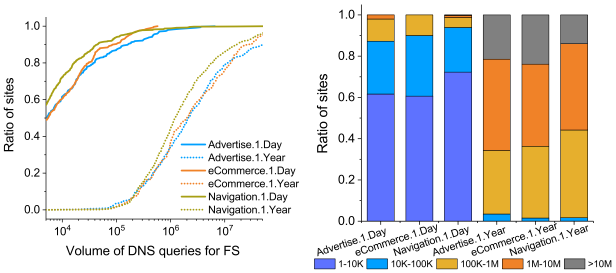


Fig. 4: Cumulative distribution of DNS query volume for FS

billion queries (53,802,674,384). The three fraud categories (eCommerce, navigation and advertising) respectively receive **2.7 billion, 38 billion and 13 billion** requests. The numbers, however, are only a lower bound of the infected population, since only ~15% of DNS traffic in China is logged by our data provider.

Furthermore, we investigated the distribution of FS requests. As illustrated in Figure 4, in the 1-year period, 35.4% FS (516) received 100K-1M DNS queries and 15.9% FS (232) received more than 10M queries. The largest volume is observed on huo99.com, which accounts for more than 19 billion requests since 05/2017 (statistics come from DNS Pai). According to a forum post[4], one kind of client-side malware always reroutes requests from Baidu and Google to huo99.com. When restricting the time window to a single day (01/10/2017 as shown in Figure 4), most FS are queried for less than 10K times (less than 1K for 15.9% FS and 1K-10K for 51.1% FS), but 2 FS receive more than 10M queries. Although fraud strategies against the three types of traffic networks vary, the distributions of request volume are similar.

Regarding the active period of domains, 1,244 FS (eCommerce: 248, navigation: 670, advertising: 326) have been visited for more than 300 days, suggesting their infrastructure is resilient against disruption.

B. Search Ad Impersonation

An adversary can buy ad space provided by search engines to trick users to visit their FS, and further defraud other parties like eCommerce sites. To check whether an FS is conducting such fraud, we need to find search ads pointing to these domains, which is infeasible when the keywords targeted by fraudsters are unknown. However, we find one Baidu's public API[5] can indicate the link between domain and search ads, as it tells whether a domain has been verified by Baidu (based on the review of legal documents) before posting search ads.

By querying the API with the 1,457 FS SLDs, we found 23 of them conducting Search Ad Impersonation. All domains automatically redirected to `taobao.com` when visited by our crawlers, suggesting they may belong to the same fraud campaign. Table V lists the top 10 FS ranked by DNS query volume. The top FS sends 314,202 requests to `taobao.com`, probably causing significant loss to the company. Later in Section VI-E, we perform a detailed analysis on the economic loss.

TABLE V: Query volume of FS in Search Ad Impersonation

Ranking	Domain Name	Query Volume
1	hao1.dambolofashion.org	314,202
2	www.svnss.com	232,153
3	www.hxfus.com	181,085
4	hao2.3506ygfs.com	180,063
5	hao2.csyycsyy.com	131,011
6	hao360.dawanbiao.cn	106,585
7	vip.wyqfc.com	91,712
8	vip.vapdj.com	89,953
9	t.qx120.net	79,901
10	vip1.meiquxian.cn	75,760

TABLE VI: Number of URLs under each FS

FS	# URL	FS	# URL
hao360.dawanbiao.cn	2,457	hao2.3506ygfs.com	660
www.hxfus.com	594	www.wlzyx.com	279
t.iavip.cn	250	vip.1314dian.cn	98
t.yunform.com	56	k.xmhex.com	41
t.wusacm.com	35	vip.srbxj.com	24

Using our URL dataset, we looked into the URLs hosted on the fraud SLDs. An interesting observation is that all URLs embed a keyword in Chinese Pinyin, which are all distinctive under one SLD. The keywords are always mapped to a search keyword priced by Baidu. As such, the number of URLs reflect the number of the search keywords targeted by an FS. Table VI lists the number of URLs observed per fraud SLD, ranked by the volume. As an example, 2,457 URLs are hosted under hao360.dawanbiao.cn. We find that fraudsters are willing to buy a large number of search keywords.

C. Transport-layer Injection

In the ground-truth dataset D_{ES}, we identified 44 FQDNs (36 SLDs) involved in transport-layer content injection. During passive analysis, 6 FQDNs (5 SLDs) detected by Content Analyzer overlap with D_{ES}. The main reason for missing the remaining FQDNs is the time gap between our empirical study and DNS data collection. A large number of URLs are invalid when visited by our crawler. The 6 FQDNs detected by `TraffickStop` are defrauding the advertising program of Baidu. For instance, the most popular URL under sta.jcjk0451.com automatically loads Baidu's ads without any other meaningful content in its web page.

Although client-side IP is anonymized in our dataset, we are able to obtain aggregated statistics about users' geo-location, using the internal threat-intelligence service of the

[4] https://zhidao.baidu.com/question/181930340.html
[5] https://www.baidu.com/s?wd=domainname@v

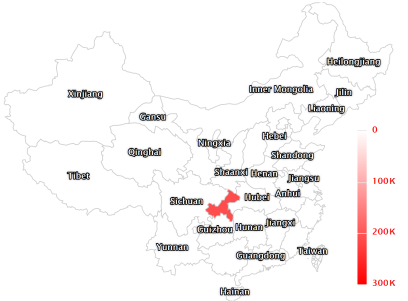


Fig. 5: Geo-distribution of `js.cqmono.cn`'s DNS queries

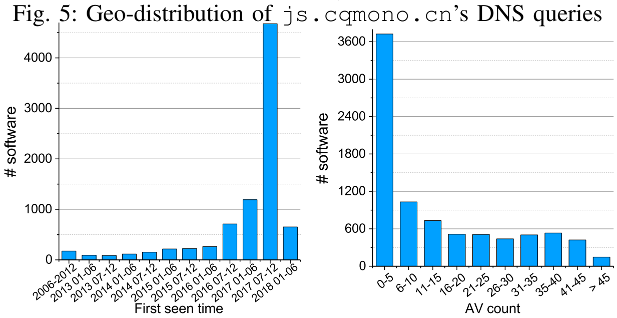


Fig. 6: First seen time and AV count of client-side softwares.

company. Surprisingly, we find that transport-layer injection is highly concentrated in a small number of regions. For instance, nearly all DNS queries (around 9.9 million in three years) going to js.cqmono.cn, one detected FS, originate from Chongqing City of China, while all other regions only see negligible queries (see Figure 5). We speculate such injection is conducted by **regional ISP**, probably at the level of city or province.

D. Client-side Hijacking

Traffic hijacking is a popular attack vector for client-side malware or grayware to profit unethically from the infected machines. We aim to gain a better understanding of their fraud strategies, which is only possible after analyzing the relevant fraud applications. To this end, we queried the sandbox run by Qihoo 360 using the fraud SLDs and logs, and discovered **36,904** applications (distinct MD5) contacting them. Then, we queried VirusTotal with the software MD5 and used the feedbacks from AVs to filter malicious ones. As a result, 8,555 are alarmed by at least one AV (31,729 have been submitted to VirusTotal). Among all the 8,555 applications, 76.2% (6,520) were submitted after 2017, but still, 4.1% (354) were discovered by the community before 2013 (see Fig 6). The detection results among AV companies vary, with about 59.9% (5,127) applications being alarmed by more than 5 AVs.

We consider the **5,127** applications alarmed by at least five AVs as fraud and further investigate them. To study their functionalities, we first classified the applications based on their AV labels, using an open-source tool named AVClass [70]. Among them, 3,063 can be classified (1,483, or 48% are PUP and the remaining 1,580 are malware), which are grouped into 341 families. Table VII lists the top 5 families and the number of applications under each. By searching their definitions and running the binaries in a sandbox environment, we find their functionalities are very diverse, including compression software, game software, cheat plugin, game cracker, key generation, and software downloader. Therefore, we suspect fraudsters choose to partner with the software authors and ask them to bundle the hijacking code. Interestingly, reflected by its volume, desktop gaming software has a much closer connection to fraudsters comparing to other software.

TABLE VII: Labeling results of client-side software (by MD5).

Family Name	Total	Malware	PUP	Description
flystudio	435	189	246	Gambling software, video downloader.
youxun	179	0	179	Game guide.
zusy	130	82	48	Game client software.
kuaizip	114	0	114	Compression software.
onlinegames	71	48	23	Game cracker, key generation.

To make it more likely to be installed, fraud software can leverage file certificates. Among the alarmed applications, 463 have certificate and 177 of them are signed by WoSign, a questionable certificate provider recently abandoned by Microsoft [22]. However, we find that VeriSign, a very reputable certificate provider, is also abused and accounts for 200 certified applications. Our discovery aligns with previous works, that file certificates are abused by PUP authors to increase the chance of installation [48].

Zooming into the behaviors of collected software, we discovered a new paradigm to profit from victim's traffic, from one kind of software named `liuliangbao` ("traffic pal" in English). One of its functionalities is to instruct the host machine to increase the "like count" and "visit count" of a designated shop in eCommerce platform, in order to gain a higher ranking. The adversary disseminates the commands (e.g., when to hit "like") in P2P fashion to avoid central failure. Detecting such fraud traffic is challenging for traffic networks, since the visits all come from real users.

E. Economic Loss

Traffic fraud inflicts loss on traffic networks when visits maneuvered by adversaries are rewarded, yet they do not bring any real value. Since we have no cooperation from the traffic networks, we estimated their probable loss based on the amount of traffic and the rewarding policies. Table VIII lists the estimated loss aggregated under several leading networks. Below we elaborate the analysis model and result.

Model. Essentially, how much an affiliate earns depends on how many visitors are referred to a PS. Since automatic redirection is enforced by adversaries and the redirection target is usually fixed, we assume all visitors coming to an FS will land on the subsequent PS. The amount of visitors is proportional to the DNS requests we observe. This estimation model is elaborated below, which is similar to a prior work studying ad fraud [60].

$$Loss_{PS}(t) = \sum_{w \in Detected} (P_{DNS}(w,t) \cdot R_{visitor}) \cdot (P_{model} \cdot F_{factor})$$

In the above equation, t is the length of the observation period and we used one day (04/01/2018) in this study. $Detected$ covers all detected FS and w are those redirecting to the designated PS. $P_{DNS}(w,t)$ is the volume of DNS queries we observe. $R_{visitor}$ is the ratio between the overall visit volume and $P_{DNS}(w,t)$. Since our dataset only covers about 15% DNS traffic in China, we set $R_{visitor}$ to 6.6[6].

[6] Previous works [41] consider domain TTL when estimating overall visits. We do not consider this factor, because our data is collected from the bottom-level recursive DNS servers and all resolution requests are captured.

TABLE VIII: Estimation of loss for the selected traffic networks.

Traffic Network	PS	# FS	DNS Volume	P_{model}	F_{factor}	Loss (per day)
eCommerce	taobao.com	150	403,964	\$ 1.346 (CPA)	0.015	\$ 53,829.8
	jd.com	64	68,661	\$ 2.785 (CPA)	0.015	\$ 18,930.9
Navigation	hao123.com	151	328,359	\$ 9.5 (per 1K IP)	0.12	\$ 2,470.6
	hao.360.cn	44	154,295	\$ 8.8 (per 1K IP)	0.12	\$ 1,075.4
Advertising	Baidu CPM	4	1,754,259	\$ 0.23 (CPM)	5	\$ 13,314.8

P_{model} is the monetary reward defined by the PS. F_{factor} estimates the conversion rate between referred visitors and visitors counted by PS. For instance, in eCommerce programs, F_{factor} measures how likely a visitor makes a purchase. However, F_{factor} per PS is usually unavailable to public. As an alternative, we filled this parameter with metrics published in public reports.

eCommerce. We choose two popular eCommerce sites in China, `taobao.com` and `jd.com` as PS, and estimate their aggregated loss. F_{factor} is set to 1.55% according to a study on billions of shopping sessions [23]. For P_{model}, we focus on the CPA (Cost Per Acquisition) rewarding model, which counts the purchases made by visitors. Given that CPA value is not fixed, depending on the commodity's price, we estimate the average reward by crawling their pages of commission listing [14] [15] and take the mean value as P_{model} after removing extreme values. \$1.346 and \$2.785 are used for `taobao.com` and `jd.com`, respectively. As a result of our analysis, `taobao` and `jd` could lose 53.8K and 18.9K dollars per day due to this fraud.

Navigation. Affiliates contracted with a navigation network are usually rewarded by the number of distinct visitor IPs. As such, we need to convert DNS requests to their associated IPs, yet our aggregated DNS dataset does not include client IPs. Consequently, we set the request-to-IP rate (i.e., F_{factor}) to 0.12 based on a pre-measurement study on DNS data of two universities (where hashes of IPs are available), meaning that one IP is responsible for 8.08 requests per day on average. We performed our loss estimation on `hao.123.com` and `hao.360.cn`, using their published rewarding terms (low bounds are \$9.5 and \$8.8 for per 1K IPs) as P_{model}. As a result, we find `hao123` and `360` could lose around 2.5K and 1.0K dollars per day due to this fraud.

Advertising. We focus on the CPM (cost per thousand impressions) model of advertising platforms, which rewards an affiliate simply by how many times an ad is displayed. We selected FS that redirects to Baidu using CPM URL (other ad URLs like CPC URLs have different patterns). An FS may display multiple ads of a PS on the same page. After inspecting all FS pages we crawl, the number is set to 5, which is used as F_{factor}. From one advertiser, we learn that Baidu's reward is \$0.23 per thousand impressions. As a result, Baidu's lost is around 13.3K dollars per day.

Remarks. Our study shows that a single program could lose thousands of dollars due to traffic fraud on each day. On the other hand, fraudsters' real profit is supposed to be less, due to the operational costs including domain registration fees, server hosting fees and others. These costs, however, can be amortized when their operation runs for a long time.

Lacking ground truth, our estimation might be inaccurate due to the following factors. First, while rare, one FS can redirect to different PS in different periods. Second, cloaking is sometimes performed so a visitor might not be forwarded to PS depending on her location and browser configuration. Still, we believe the fraud issue revealed by our study is clear, which should be taken serious consideration by traffic networks.

TABLE IX: Publishers reselling ads to FS

Publisher	Alexa Ranking	Evidence (redirection chain)
`Publisher-1`	~ 200	http://hao.67it.com:86/dfadtz023.js http://mini.e*s*d*y.com/?qid=sytest23 http://dup.b*i*u*t*t*c.com/js/ds.js
`Publisher-2`	~ 1000	http://t.5txs.cn/rb/i9.js http://11.m*d*i*e*s.com/****/baiduAfxId.html http://www.d***.com/union2.html?u207 http://cpro.b*i*u*t*t*c.com/cpro/ui/c.js
`Publisher-3`	~ 4000	http://m.cnepin.cn/cl/html/jd34.html http://bj.g****.com/content/contentbranch.php? http://cpro.b*i*u*t*t*c.com/cpro/ui/c.js

F. Ad Reselling

Advertising platforms usually perform reverse-crawling of publisher sites to determine whether they commit fraud. To evade the crawling-based detection, some fraudsters choose to participate in *ad reselling* programs of other legitimate publishers and earn a share by sending hijacked traffic. Assume a publisher (say P) embeds ads from an advertiser (say A). To increase revenue, P allows other sites (say F) to load its ads. A portion of the reward from A will be distributed to F. Although F earns less than directly contracting with A, it is safer as A only sees P's URL.

We find 7 companies in China (3 are very popular) resell ads from Baidu advertising platform to FS. Those gray publishers are captured by our Association Finder and confirmed during dynamic crawling. Table IX shows the browser redirection chain from FS to grey publishers and finally PS[7]. Surprisingly, we find that Publisher-2, a famous anti-virus site in China (Alexa ranking 1K), resells its Baidu ads to other fraud sites, like 5txs.cn which shows content violating Baidu's policies.

By reviewing the content of websites, advertising platforms try to avoid being abused by unfriendly adversaries, such as porn and lottery websites. However, ad reselling makes this attempt ineffective, and we suggest the platforms revisit their review process.

G. Domain Renting

We find that many owners of fraud SLDs involve *domain renting*, a strategy that pays a domain owner to rent her registered domain for a short period. During the lease, tenants are allowed to point the rented domain to any server. Such

[7] We anonymized the famous publishers found to be participating in ad reselling due to legal restrictions imposed by Tsinghua University and Netlab of 360.

practice is popular in China's domain market, as it saves the tenant from a lengthy procedure of getting an ICP license (described later). It could also completely evade state-of-the-art detection systems relying on domain reputation: none of the registration fields are changed before and after lease. Therefore, approaches leveraging creation time [42] or change of registrant [51], [54] will be ineffective. Below, we first describe the concept of ICP license and then report our findings on how domain renting is abused. To better understand this issue, we perform an infiltration study and report our results.

ICP license. According to regulations of domain name management in China, websites with business purposes must be reviewed by the government department before providing services. When all checks are passed, an Internet Content Provider (ICP) license is issued [12], [17]. The comprehensive review process requires valid ID cards and photos from applicants, which becomes an obstacle for cyber-criminals. As such, instead of applying by themselves, adversaries have strong incentive to obtain ICP licenses from others. Moreover, revoking ICP licenses is complicated as well when the owner decides to terminate the site, where ID and filled forms are needed again. Therefore, ICP license usually keeps being attached to the domain, even when its owner changes.

Abuse of domain renting. Due to the issues discussed above, domains with valid ICP licenses are in strong need in China. Many companies and persons have started to grasp such domains and rent them for profit. Those entities are called "MiNong" (in Chinese Pinyin, meaning "rice farmers") in China, and their main approach is drop-catch registration [43], [51] (i.e., registering a domain as soon as it expires). After owning the domain, MiNong publishes the domain name in advertisement and looks for tenants. The checks performed by MiNong when transferring the domain are quite loose and there is no regulation about how the tenant uses the domain. By correlating ICP data of domains[8], such domains can be identified: if the registration date of a domain is *later* than the issue date of its ICP license[9], it is a MiNong domain.

We inspected the 1,457 fraud SLDs using this rule, and discovered that 74 are rented from MiNong (eCommerce:11, advertising:37, navigation:26). We further applied this rule on a larger dataset of 188,967 SLDs identified by Association Finder, and found 6,865 (6.81%)[10] are rented from MiNong. By clustering the SLDs by registrant's email, we find most of them are controlled by a dozen of registrants (see Table X). All personal email addresses point to professional domain dealers who offer domain renting. The remaining are three domain proxy companies (juming.com, idczh.com and hichina.com), who claim to offer domains with valid ICP license [5], [6] and that the domain names are able to bypass the government's review process.

Lifetime before MiNong. Next, we measured the duration from the initial domain registration (which is also approximately the issue date of ICP license) till acquired by MiNong (the registration date after). The result shows that the duration for 78.29% (5,375) domains is longer than one year and more than 5 years for 7.3% (503) domains. Given that reputation-based systems tend to assign high suspicious scores to domains used for less than 1 year [26], capturing domains rented from MiNong with long lifetime would be more difficult.

TABLE X: WHOIS Email information of the 6,865 SLDs found by Association Finder.

# Domain	Email	MiNong
1,964	WHOIS privacy protection[a]	Not Sure
1,311	yaomaiyumingzhaowo@126.com	✓
585	admin@juming.com	Providing Service
392	dt0598@outlook.com	✓
316	23362464@qq.com	✓
262	apple4407@163.com	✓
208	80010864@qq.com	✓
175	domain@idczh.com	Providing Service
125	8648240@qq.com	✓
104	2893741234@163.com	✓
102	jubaociyuming@126.com	✓
96	domainadm@hichina.com	Providing Service

[a] Registration proxy services like `yinsibaohu.aliyun.com` hide registrants' identities.

Infiltration study. To better understand the business model of MiNong, we contacted one through ICQ pretending to be a buyer, and were given a list of domains with valid ICP licenses. After we ordered one, babybride.cn, the admin credentials on `DNSPOD` (a domain management platform run by Tencent) were transfered to us. Throughout the process, all registration data of this domain remained unchanged, although the domain has been rented to us. Renting the domain for 6 months costs $7.35, which is more expensive than directly buying the domain ($5.14 per year from DNSPOD).

Meanwhile, we performed a study on how MiNong deals with expired domains. In particular, we registered a domain (msgiraffe.com) and obtained an ICP license after the review process. The domain expired on 02/22/2017 and was immediately registered by whoisagent@hkDNS.hk (also providing domain renting service) *on the same day*. We speculate our domain has been monitored by MiNong after the registration, as ICP information is publicly available. Later in June 2017, this domain was alarmed by Tencent Security due to its use for fraud.

H. Case Study

We show how adversaries can combine the traffic manipulation and evasion strategies in one campaign with an example.

`TraffickStop` detects one FS, `r9.5txs.cn`, which is used by fraudsters to defraud Baidu advertising networks. While a redirecting script is found on this domain, its SLD (`5txs.cn`) cannot be resolved. Instead of directly loading Baidu ads, it chooses to load ads from `Publisher-2` and `Publisher-3`, which resell their Baidu ads for profit (see Section VI-F). In addition, the domain is rented from a MiNong so it is difficult to identify its real owner. When the domain is plugged into the fraud campaign, a steep spike of DNS queries is observed, as shown in Figure 7. More than 220K requests are seen on 07/10/2016 but no request is observed on the prior day. Furthermore, we also find that the domain was accessed by `liuliangbao`, a client-side PUP (see Section VI-D). We speculate that `liuliangbao` hijacks users' traffic to `r9.5txs.cn` for ad fraud, which is only one of its tasks, as it is also found to attack other traffic networks like Taobao's eCommerce network.

[8] provided by http://icp.chinaz.com

[9] The ICP review process requires a domain being registered first.

[10] We only obtained WHOIS record for 100,742 domains.

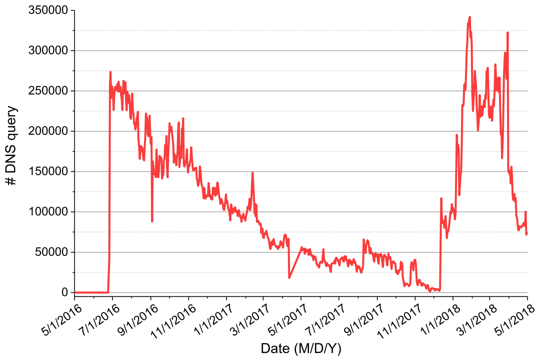


Fig. 7: DNS queries of `*.5txs.cn` seen by passive DNS.

VII. Discussion

Browser multi-tasking. Web requests are not always sequentially issued due to the optimization of modern browsers. Multiple tabs could be opened and multiple web files could be loaded simultaneously. Many "noisy" domain requests would be injected between the visit of FS and PS, making FS fall out of the sliding window. In addition, the DNS prefetching feature [34] allows DNS resolutions to be completed in a batch before clicking the associated hyperlinks, causing similar results. To address this issue, we set a large window size (40 domains) and correlation threshold based on the preliminary measurement on two universities' datasets to offset the impact of the "noisy" domains (see Appendix B).

DNS caching. The resolution result of a domain could be cached by the stub resolver. Since PS is supposed to be visited more frequently than FS and more likely to be cached, the support value could be diminished. As a result, it makes FS harder to be discovered by *Associate Finder*. We choose a small threshold *minconf* considering this factor (see Appendix B).

Evasion. Knowing the design of `TraffickStop`, adversaries can manipulate the redirection behaviors or page content in order to evade detection. However, making changes along these directions would inevitably introduce negative impact to the fraud campaigns. First, to reduce the degree of correlation with PS, FS could request many URLs from other websites before redirecting to PS, or asking visitors to click the hyperlink of PS. However, the approach would require more waiting time of visitors and likely turn them away. Second, FS directly accessed by IP is not captured by `TraffickStop`, but switching to IP could easily expose the geo-location of FS and its hosting services. In addition, a URL without a domain name is more suspicious, which has been considered as an indicator in previous works [58]. Third, when adversaries host FS on CDN services and use FQDN under the CDN SLD, domain folding performed during our data sorting stage would merge the FQDN with other innocent ones. As a countermeasure, domain folding is not performed on FQDNs of known CDN services (see Section IV-B). Forth, while adversaries can give each FS a unique look and structure in order to evade our clustering algorithm, maintaining randomized code is much more difficult. As such, web pages generated from the fixed template is preferred. The same observation is also documented by previous studies [52], [55]. We acknowledge that code randomization could bypass our current detection. However, the core of our detection system is generating "potential" fraud domain names from the perspective of large-scale DNS analysis. We develop the content analyzer based on HTML DOM structure, which is simple and straightforward, to evaluate how the idea works and understand the strategies of illicit traffic monetization. Our content analyzer only provides a baseline of the scale of fraud websites for our measurement study. As future work, advanced content analyzer can be developed to identify complex fraud clusters.

Generality of our study. `TraffickStop` requires finer-grained DNS data to identify FS. Our industrial partner provides us with the access to its central database, which collects logs from DNS resolvers mainly in China. While there are coarse-grained passive DNS datasets open to public, we have not obtained access to another finer-grained DNS dataset. Meanwhile, *the three types of monetization are not China-specific* [32], [53], [61], [71], [72] and *our detection features are general*. As such, we believe `TraffickStop` can capture illicit traffic monetization in other regions when applying on different DNS data. Although not verified through rigorous analysis, we speculate this fraud activity is more prevalent in China, due to the slow progress of HTTPS deployment [39].

VIII. Related Work

Advertisement fraud. Previous studies have shown that fraud against advertising companies can cause huge loss. How to detect ad fraud is a hot topic and prior works either analyze data collected on advertiser side [30], [31], [32], [72] or scan client-side applications [47], [63], [74], [77]. By contrast, `TraffickStop` is able to detect ad fraud conducted by adversaries attacking any advertiser site at any level (client-side, transport-layer and server-side), rather than protecting just one ad network (*e.g.*, [31]) or one ecosystem [74]. This is achieved through a novel approach on DNS analysis, which has never been explored before.

Fraud in affiliate marketing. In addition to ad fraud, `TraffickStop` also identifies fraud activities against eCommerce Network (or affiliate marketing). Edelman *et al.* elaborate its business model [36]. This topic gains traction from the academia recently. One popular fraud technique is "cookie stuffing": adversaries overwrite the cookie set by legitimate publishers and claim the sale on their behalf. Chachra *et al.* study this issue by crawling a large volume of vulnerable publishers [27]. Snyder *et al.* propose an approach to detecting such fraud by analyzing HTTP request logs passively [71]. In this paper, we identify another type of affiliate fraud which is conducted through Search Ad Impersonation, and our study complements existing researches regarding this threat.

Content injection by network adversaries. Our study detects malicious code injected into users' browsing sessions by network adversaries (*e.g.*, ISP, in-path devices). Such adversary model has been known for long. Reis *et al.* develop Web Tripwires [68] to help site owners detect in-flight modifications. Dagon *et al.* [29] and Kührer *et al.* [49] examine millions of DNS resolvers through active probing and find a considerable amount of them are injected with false content, like unwanted ads or even malicious code. Liu *et al.*studies the hidden DNS traffic interception performed by network middlebox recently [57]. A recent work by Nakibly *et al.* studies the out-of-band content injection performed by both edge (ISP) and non-edge network operators [61]. They deploy a monitoring system at the entry points of large networks and search for out-of-band sessions. Though effective, their systems require cooperation from website owners and network operators or require frequent Internet scans. As such, this schema cannot be deployed at a larger scale. Instead, our system spots the anomalies from passive DNS logs, without incurring any deployment burden to those parties.

Passive DNS analysis. Many DNS providers have participated in programs that allow access to their DNS logs, which prompts many security applications built on top. For instance, passive DNS data has been utilized to detect malicious domains [24], [25], [26], [66], assess the population affected by drive-by-download exploit kit [41], and examine the behaviors of DNS resolvers globally [40]. We demonstrate that the same data can be leveraged to detect traffic fraud as well.

IX. Conclusion

In this paper, we present a new detection system against illicit traffic monetization. Identifying the artifacts of fraud infrastructure, i.e., FS, is by no means trivial, but we are able to achieve this goal through novel correlation analysis on DNS data. In the end, our system detects 1,457 FS set up against eCommerce, advertising and navigation networks, with an accuracy of 72.7%. We have reported the detection results to relevant companies and received acknowledgements. Starting from the FS, we investigate various factors of this fraud business, including the scale, evasion strategies and impact on legitimate parties. Our results show the substantial impact of this fraud and difficulties in entire mitigation. Meanwhile, our detection features are general, we believe our system can also work in other regions when applying on finer-grained DNS data. We hope the features and insights gained through this study could spur new approaches tackling this problem.

Acknowledgments

We thank all anonymous reviewers for their helpful suggestions and comments to improve the paper. This work was supported by the National Key Basic Research Program (grant 2017YFB0803202), the National Natural Science Foundation of China (grant 61772307, U1836213, U1636204), and CERNET Innovation Project NGII20160403. The IU author is supported in part by the NSF CNS-1527141, 1618493, 1801432, 1838083 and ARO W911NF1610127. Any views, opinions, findings, recommendations, or conclusions contained or expressed herein are those of the authors, and do not necessarily reflect the position, official policies or endorsements, either expressed or implied, of the Government of China or Qihoo 360.

References

[1] Agreements of 360 navigation affiliate service. http://lianmeng.360.cn/hao360_agreement.html.

[2] Agreements of Alimama affiliate programs. http://help.alimama.com/?spm=a2321.7393629.1998051879.3.7dJicq#!/u/faq/detail?id=5704248.

[3] Agreements of Baidu affiliate programs. http://yingxiao.baidu.com/union/detail_5264.html.

[4] Baidu's website hidden malicious code and stealing traffic. http://tech.sina.com.cn/i/2017-02-28/doc-ifyavvsk3974317.shtml.

[5] Booking domain name with ICP license. http://whois.hkdns.hk/club/thread-45803-1-1.html.

[6] Booking domain names. http://www.idczh.com/icp.asp.

[7] Browser hijacking. http://bbs.kafan.cn/thread-2037071-1-1.html.

[8] Browser hijacking. http://bbs.kafan.cn/thread-2041256-1-1.html.

[9] Description of Hao123 PC affiliate. http://yingxiao.baidu.com/zhichi/knowledge/detail.action?channelId=16&classId=11630&knowledgeId=11638.

[10] Description of Hao123 pricing models. http://yingxiao.baidu.com/zhichi/knowledge/list.action?pid=11959&channelId=16&classId=11630.

[11] Description of Hao123 Wireless affiliate. http://yingxiao.baidu.com/zhichi/knowledge/list.action?pid=13414&channelId=16&classId=13414.

[12] Law of non-profit internet information service. http://www.gov.cn/gongbao/content/2005/content_93018.htm.

[13] The list of CDN. https://zenodo.org/record/842988#.WmKUPZM-dMF.

[14] The list of CPS. https://pub.alimama.com/promo/item/channel/index.htm?channel=qqhd.

[15] The list of CPS. https://media.jd.com/gotoadv/goods?pageSize=50.

[16] Malicious sites are related with browser hijacking. https://www.zhihu.com/question/21883209.

[17] Management law of internet information service. http://www.gov.cn/gongbao/content/2000/content_60531.htm.

[18] News of Taobao affiliate. https://www.alimama.com/news_detail.htm?contentId=1246.

[19] Passive DNS system. http://www.passivedns.cn.

[20] Public suffix list. https://publicsuffix.org/.

[21] A survey study about traffic hijacking in china. http://www.freebuf.com/articles/web/104426.html.

[22] Microsoft disables support for chinese digital certificate providers wosign, startcom. http://gadgets.ndtv.com/apps/news/microsoft-wosign-startcom-china-digital-certificate-providers-revokes-window-10-support-1735848, 2017.

[23] Mobile marketing. http://www.smartinsights.com/mobile-marketing/mobile-commerce/mobile-users-still-not-converting, 2017.

[24] Antonakakis, M., Perdisci, R., Dagon, D., Lee, W., and Feamster, N. Building a dynamic reputation system for dns. In *USENIX Security* (2010).

[25] Antonakakis, M., Perdisci, R., Lee, W., Vasiloglou, II, N., and Dagon, D. Detecting malware domains at the upper DNS hierarchy. In *USENIX Security* (2011).

[26] Bilge, L., Kirda, E., Christopher, K., and Balduzzi, M. EXPOSURE: Finding malicious domains using passive DNS analysis. In *NDSS* (2011).

[27] Chachra, N., Savage, S., and Voelker, G. M. Affiliate crookies: Characterizing affiliate marketing abuse. In *IMC* (2015).

[28] Chen, J., Jiang, J., Duan, H., Weaver, N., Wan, T., and Paxson, V. Host of troubles: Multiple host ambiguities in http implementations. In *IMC* (2016).

[29] Dagon, D., Lee, C., Lee, W., and Provos, N. Corrupted dns resolution paths: The rise of a malicious resolution authority. In *NDSS* (2008).

[30] Dave, V., Guha, S., and Zhang, Y. Measuring and fingerprinting click-spam in ad networks. *ACM SIGCOMM Computer Communication Review* (2012).

[31] Dave, V., Guha, S., and Zhang, Y. Viceroi: Catching click-spam in search ad networks. In *ACM SIGSAC* (2013).

[32] DeBlasio, J., Guha, S., Voelker, G. M., and Snoeren, A. C. Exploring the dynamics of search advertiser fraud. In *Proceedings of the 2017 Internet Measurement Conference* (2017), ACM, pp. 157–170.

[33] DNS Pai. http://www.dnspai.com.

[34] DNS-Prefetching. https://dev.chromium.org/developers/design-documents/dns-prefetching, 2017.

[35] Du, K., Yang, H., Li, Z., Duan, H.-X., and Zhang, K. The ever-changing labyrinth: A large-scale analysis of wildcard dns powered blackhat seo. In *USENIX Security Symposium* (2016), pp. 245–262.

[36] Edelman, B., and Brandi, W. Risk, information, and incentives in online affiliate marketing. *Journal of Marketing Research* (2015).

[37] Englehardt, S., Eubank, C., Zimmerman, P., Reisman, D., and Narayanan, A. Openwpm: An automated platform for web privacy measurement. In *CCS* (2016).

[38] FarSight. https://www.farsightsecurity.com/solutions/dnsdb.

[39] Felt, A. P., Barnes, R., King, A., Palmer, C., Bentzel, C., and Tabriz, P. Measuring https adoption on the web. In *USENIX Security* (2017).

[40] Gao, H., Yegneswaran, V., Chen, Y., Porras, P., Ghosh, S., Jiang, J., and Duan, H. An empirical reexamination of global dns behavior. In *SIGCOMM* (2013).

[41] Grier, C., Ballard, L., Caballero, J., Chachra, N., Dietrich, C. J., Levchenko, K., Mavrommatis, P., McCoy, D., Nappa, A., Pitsillidis, A., et al. Manufacturing compromise: the emergence of exploit-as-a-service. In *CCS* (2012).

[42] Hao, S., Kantchelian, A., Miller, B., Paxson, V., and Feamster, N. PREDATOR: Proactive recognition and elimination of domain abuse at time-of-registration. In *CCS* (2016).

[43] Hao, S., Thomas, M., Paxson, V., Feamster, N., Kreibich, C., Grier, C., and Hollenbeck, S. Understanding the domain registration behavior of spammers. In *IMC* (2013).

[44] Harrington, P. *Machine learning in action*. Manning Greenwich, CT, 2012.

[45] Heaton, J. Comparing dataset characteristics that favor the apriori, eclat or fp-growth frequent itemset mining algorithms. In *SoutheastCon* (2016).

[46] ICANN. New icann registrant update requirements. https://www.godaddy.com/help/new-icann-registrant-update-requirements-24654.

[47] Kapravelos, A., Grier, C., Chachra, N., Kruegel, C., Vigna, G., and Paxson, V. Hulk: Eliciting malicious behavior in browser extensions. In *USENIX Security 14* (2014).

[48] Kotzias, P., Bilge, L., and Caballero, J. Measuring pup prevalence and pup distribution through pay-per-install services. In *USENIX Security Symposium* (2016), pp. 739–756.

[49] Kührer, M., Hupperich, T., Bushart, J., Rossow, C., and Holz, T. Going wild: Large-scale classification of open dns resolvers. In *IMC* (2015).

[50] Kwon, B. J., Mondal, J., Jang, J., Bilge, L., and Dumitras, T. The dropper effect: Insights into malware distribution with downloader graph analytics. In *CCS* (2015).

[51] Lauinger, T., Chaabane, A., Buyukkayhan, A. S., Onarlioglu, K., and Robertson, W. Game of registrars: An empirical analysis of post-expiration domain name takeovers. In *USENIX Security 17* (2017).

[52] Levchenko, K., Pitsillidis, A., Chachra, N., Enright, B., Félegyházi, M., Grier, C., Halvorson, T., Kanich, C., Kreibich, C., Liu, H., McCoy, D., Weaver, N., Paxson, V., Voelker, G. M., and Savage, S. Click trajectories: End-to-end analysis of the spam value chain. In *Security and Privacy* (2011).

[53] Lever, C., Kotzias, P., Balzarotti, D., Caballero, J., and Antonakakis, M. A lustrum of malware network communication: Evolution and insights. In *Security and Privacy (SP), 2017 IEEE Symposium on* (2017), IEEE, pp. 788–804.

[54] Lever, C., Walls, R. J., Nadji, Y., Dagon, D., McDaniel, P. D., and Antonakakis, M. Domain-z: 28 registrations later measuring the exploitation of residual trust in domains. In *Symposium on Security and Privacy* (2016).

[55] Li, Z., Alrwais, S., Xie, Y., Yu, F., and Wang, X. Finding the linchpins of the dark web: a study on topologically dedicated hosts on malicious web infrastructures. In *Security and Privacy* (2013).

[56] Lian, W., Rescorla, E., Shacham, H., and Savage, S. Measuring the practical impact of dnssec deployment. In *USENIX* (2013).

[57] Liu, B., Lu, C., Duan, H., Liu, Y., Li, Z., Hao, S., and Yang, M. Who is answering my queries: Understanding and characterizing interception of the {DNS} resolution path. In *27th {USENIX} Security Symposium ({USENIX} Security 18)* (2018), pp. 1113–1128.

[58] Ma, J., Saul, L. K., Savage, S., and Voelker, G. M. Beyond blacklists: learning to detect malicious web sites from suspicious urls. In *In SIGKDD* (2009), ACM, pp. 1245–1254.

[59] Mirsky, Y., Doitshman, T., Elovici, Y., and Shabtai, A. Kitsune: An ensemble of autoencoders for online network intrusion detection. In *NDSS* (2018).

[60] Moore, T., Leontiadis, N., and Christin, N. Fashion crimes: trending-term exploitation on the web. In *CCS* (2011).

[61] Nakibly, G., Schcolnik, J., and Rubin, Y. Website-targeted false content injection by network operators. In *USENIX Security 16* (2016).

[62] Paxson, V., Christodorescu, M., Javed, M., Rao, J. R., Sailer, R., Schales, D. L., Stoecklin, M. P., Thomas, K., Venema, W., and Weaver, N. Practical comprehensive bounds on surreptitious communication over dns. In *USENIX Security* (2013).

[63] Pearce, P., Dave, V., Grier, C., Levchenko, K., Guha, S., McCoy, D., Paxson, V., Savage, S., and Voelker, G. M. Characterizing large-scale click fraud in zeroaccess. In *ACM SIGSAC* (2014).

[64] PIR. Zone file access for .org. https://pir.org/resources/file-zone-access/.

[65] PPC-fraud. http://searchengineland.com/ppc-fraud-ring-impersonated-300-advertisers-may-2014-192801, 2014.

[66] Rahbarini, B., Perdisci, R., and Antonakakis, M. Segugio: Efficient behavior-based tracking of malware-control domains in large isp networks. In *DSN* (2015).

[67] Rakuten-LinkShare. https://rakutenmarketing.com/affiliate.html.

[68] Reis, C., Gribble, S. D., Kohno, T., and Weaver, N. C. Detecting in-flight page changes with web tripwires. In *NSDI* (2008).

[69] Scrapy. https://scrapy.org.

[70] Sebastián, M., Rivera, R., Kotzias, P., and Caballero, J. AVClass: A Tool for Massive Malware Labeling. In *Symposium on Research in Attacks, Intrusions and Defenses* (2016).

[71] Snyder, P., and Kanich, C. No please, after you: Detecting fraud in affiliate marketing networks. In *WEIS* (2015).

[72] Springborn, K., and Barford, P. Impression fraud in on-line advertising via pay-per-view networks. In *USENIX Security 13* (2013).

[73] Symantec. Worldwide intelligence network environment. http://http://securityresponse.symantec.com/about/profile/universityresearch/sharing.jsp.

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

[75] validation rate, A. https://stats.labs.apnic.net/dnssec.

[76] Versign. Top-level domain zone file information. https://www.verisign.com/en_US/channel-resources/domain-registry-products/zone-file/index.xhtml.

[77] Xing, X., Meng, W., Lee, B., Weinsberg, U., Sheth, A., Perdisci, R., and Lee, W. Understanding malvertising through ad-injecting browser extensions. In *WWW* (2015).

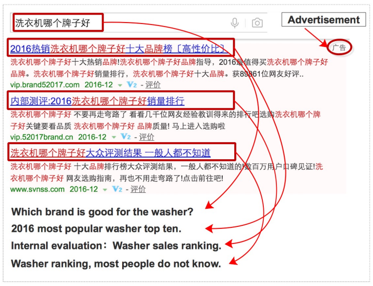
[78] Zhang, Q., Wang, D. Y., and Voelker, G. M. Dspin: Detecting automatically spun content on the web. In *NDSS* (2014).

Appendix

A. Example of Fraud Search Ad

Figure 8 shows the search results of Baidu when a user queries long-tail keywords about washers (in Chinese). Even though the ads all take visitors to tmall.com after several redirections, none of them belongs to tmall. In the meantime, fraudsters register themselves as affiliates of alimama, an affiliate program of tmall. Buying ad spaces costs money from fraudsters, but they are able to profit through pricing arbitrage when visits are converted to sales, given that the commission reward from alimama is much higher.

Such practice is forbidden by many eCommerce networks [2], [18]. Firstly, the eCommerce site also serves search ads and it has to compete against fake ads, driving up the campaign cost. In addition, payment to such affiliates for



洗衣机哪个牌子好

[2016热销洗衣机哪个牌子好十大品牌榜〔高性价比〕](#)
洗衣机哪个牌子好十大热销品牌!洗衣机哪个牌子好品牌指导，2016最值得买洗衣机哪个牌子好品牌。洗衣机哪个牌子好销量排行，洗衣机哪个牌子好十大品牌。获89861位网友好评..
vip.brand52017.com 2016-12 V2 - 评价

[内部测评:2016洗衣机哪个牌子好销量排行](#)
洗衣机哪个牌子好 不要再走弯路了 看看几千位网友经验教训得来的排行吧选购洗衣机哪个牌子好关键要看品质 洗衣机哪个牌子好 品牌质量! 马上进入选购啦
vip.52017brand.cn 2016-12 V2 - 评价

[洗衣机哪个牌子好大众评测结果 一般人都不知道](#)
洗衣机哪个牌子好 十大品牌排行榜大众评测结果，一般人都不知道的!数百万用户口碑见证!洗衣机哪个牌子好 网友选购指南，再也不用走弯路了!点击前往吧!
www.svnss.com 2016-12 V2 - 评价

Which brand is good for the washer?
2016 most popular washer top ten.
Internal evaluation：Washer sales ranking.
Washer ranking, most people do not know.

Fig. 8: Three fake search ads placed by fraudsters in Baidu search results. Notice that none of their hosting domains belongs to Tmall, though ultimately the redirection leads to `tmall.com`.

commission is unnecessary, when authentic search ads can lead to the same product page.

B. *Parameter Settings*

Client IP threshold. We want to reduce negative influences brought by NAT and proxy IPs. This is because IPs with a large number of distinct FQDNs may represent many users, e.g., NAT IP or proxy IP. Under this circumstance, the DNS sessions of multiple users are merged, lowering the accuracy of *Association Finder*.

In the end, we filtered a set of DNS requests sent by a group of abnormal client IPs based on the number of FQDNs visited per day. We inspected our university DNS dataset, among which there is no NAT or proxy IP, and find that the average number of FQNDs sent by one client per day is several thousand. We set the threshold to 100K to remove client IP visiting abnormal amount of FQNDs. As a result, 2,597 client IPs were pruned.

Sliding window size. The redirection by FS usually ends in a short interval: all sites in our ground-truth dataset finish redirection within 10 seconds on our desktop machines. We could set the sliding window to 10 seconds, but comparing timestamps is a costly operation for Association Finder. Instead, we use the number of visited domains to simulate this interval. We chose 40 domains as the window size after measuring the relation between time elapse and DNS requests on the university datasets. Choosing a larger window size will incur higher performance penalty, though more suspicious pairs might be identified.

minsup and minconf. Similarly, we examined the university datasets to select thresholds that can adequately model strong correlation. Specifically, we selected sites automatically loading JavaScript from `cnzz.com`, a popular web analytics in China, and discover 3,021 in total. We sampled some sites with strong correlation to `cnzz.com` (i.e., `cnzz.com` is the sole redirection target), and find the confidence values are usually similar when the support values are close. Therefore, we can select the support threshold based on the distribution of confidence values. When the support values are larger than 100 (1,952 sites), the confidence values are mostly above 0.1. When the support values are between 50 to 100, the confidence values are mostly above 0.2. For the remaining, the confidence values are much more random. Therefore, we use two pairs of $minsup$ and $minconf$, $\langle 100, 0.1\rangle$ and $\langle 50, 0.2\rangle$.

We want to use a small $minsup$ because our DNS data is only a small sample of the entire DNS traffic from China. $minconf$ is also small due to DNS caching: PS like `baidu.com` are likely to be cached by client machine before FS are visited, making some of the relevant requests missed by DNS Pai dataset.