ViceROI: Catching Click-Spam in Search Ad Networks

Vacha Dave*
UC San Diego
vacha@cs.ucsd.edu

Saikat Guha
Microsoft Research India
saikat@microsoft.com

Yin Zhang
Univ. of Texas at Austin
yzhang@cs.utexas.edu

ABSTRACT

Click-spam in online advertising, where unethical publishers use malware or trick users into clicking ads, siphons off hundreds of millions of advertiser dollars meant to support free websites and apps. Ad networks today, sadly, rely primarily on security through obscurity to defend against click-spam. In this paper, we present Viceroi, a principled approach to catching click-spam in search ad networks. It is designed based on the intuition that click-spam is a profit-making business that needs to deliver higher return on investment (ROI) for click-spammers than other (ethical) business models to offset the risk of getting caught. Viceroi operates at the ad network where it has visibility into all ad clicks. Working with a large real-world ad network, we find that the simple-yet-general Viceroi approach catches over six very different classes of click-spam attacks (e.g., malware-driven, search-hijacking, arbitrage) without any tuning knobs.

Categories and Subject Descriptors

K.6.5 [**Management of Computing and Information Systems**]: Security and Protection—*Advertising Fraud*

Keywords

Click-Spam, Click-Fraud, Invalid Clicks, Traffic Quality

1. INTRODUCTION

Background and motivation. Click-spam in online advertising, where unethical publishers[1] trick users into clicking ads or use malware to click on ads, hurts the online economy by siphoning off millions of advertiser dollars meant to support free websites and apps [27]. Reputed ad networks[2] attempt to filter click-spam to increase advertisers' confidence in their network [29]. Click-spam, however, is an arms-race and attacks have evolved to avoid detection [4].

Ad networks today filter click-spam *reactively* and in an *ad-hoc* manner — when a specific attack is detected (often by the impacted advertiser), the ad networks creates a filter tuned to the detected attack [29]. For example, if an advertiser complains getting thousands of clicks from a single IP address none of which convert into paying customers, the ad network may start filtering all clicks from that specific IP address (or that /24 subnet). Reactive filtering harms advertisers since attacks may go undetected for months; in one case it was estimated that click-spammers siphoned off at least $14 million over 4 years before being taken down [5]. Furthermore, ad-hoc point-solutions are quickly circumvented by attackers, e.g., avoiding the IP blacklist by using a distributed botnet, potentially adding months before the attack is rediscovered by a more savvy advertiser. A controlled measurement study conducted in 2012 found that major ad networks still missed ongoing click-spam attacks that accounted for an estimated 10–25% of clicks in the study [4].

Ad networks recognize their point-solutions to be weak and rely primarily on *security through obscurity* for protection — they fiercely guard their filtering techniques in fear that "unethical [parties] will immediately take advantage of this information to conduct more sophisticated fraudulent activities undetectable by [the ad network]'s methods" [29]. The evolution of click-spam malware, however, demonstrates the futility of relying primarily on security through obscurity. The TDL4 botnet for instance, which is estimated to have siphoned millions, avoids threshold based filters by performing only one click per IP address per day (but doing so from millions of bots), avoids browser signature based filters by plugging into real browsers, and avoids user behavior based filters by gating malware actions on user actions [4].

Approach and contributions. We had three goals in designing Viceroi: 1) *proactively* filter click-spam attacks, 2) in a way that even after we *publicly disclose* our approach it is hard (but perhaps not impossible) for click-spammers to circumvent, and 3) *simple and performant* enough that it can be deployed at Internet scales. Briefly, Viceroi meets these three goals as follows (Section 4 presents design details).

First, proactive filtering requires a very *general* approach that makes no assumptions about the specific attack mechanism. Thus as the click-spammer evolves the attack mechanisms over time, the basic filtering approach remains unaffected. A test of whether an approach is general enough in practice is to see the diversity of attacks the approach can

*Work done while at Microsoft Research India

[1]Publishers are websites, apps, or games that show ads in exchange for a fraction of the revenue generated by ad clicks.

[2]Ad networks aggregate ads from advertisers and broker them to publishers, e.g., Google AdSense, Bing Ads, Baidu.

CCS'13, November 4–8, 2013, Berlin, Germany.

http://dx.doi.org/10.1145/2508859.2516688.

detect without any tuning parameters. Viceroi detected six very different classes of ongoing click-spam attacks — including malware-driven, search-hijacking, arbitrage, conversion-fraud, ad-injection, and parked-domains — without any tuning knobs. Section 6 presents detailed case-studies.

Second, publicly disclosing the Viceroi approach without weakening it (i.e., rejecting the flawed ad networks' practice of security primarily through obscurity) requires us to focus on *invariants* — something the click-spammer cannot easily change without undermining his business model. Viceroi is designed around a simple invariant that we identified — that a click-spam attack must have higher return-on-investment (ROI) for the click-spammer than a ethical publisher to offset the risk of getting caught. Viceroi, in essence, flags publishers with anomalously high ROI. While publisher ROI is hard to estimate, in practice we found per-user revenue a close proxy. To avoid detection by Viceroi, click-spammers must reduce their per-user revenue to that of an ethical publisher. At which point, without the economic incentive to offset the risk of getting caught (by approaches complementing Viceroi), the net effect is a disincentive to commit click-spam.

Finally, to operate at Internet scales, Viceroi must deal with massive volumes of noisy ad network data efficiently and with low false-positives. Viceroi has very good performance on ROC and precision-recall curves (around 90% typically). Furthermore, in simulated attacks, Viceroi has withstood attacks against adversaries several times more powerful than the ones known today.

We believe Viceroi meets all three goals as evidenced by a large ad network deploying several aspects of our approach.

2. TERMINOLOGY

At it's simplest, online search advertising has three players; *advertisers* who want to advertise a product or service, *publishers* that run websites (search engines, news sites, blog sites), mobile apps and games that display the ads, and *ad networks* (like Google AdSense, Bing Ads, Baidu, and Yahoo) that connect advertisers with the publishers. There are two kinds of publishers: publishers owned and operated (O&O) by the ad network, e.g., google.com shows ads from Google's ad network, and syndicated publishers not controlled by the ad network, e.g., ask.com is a syndicated publisher for Google ads.

Cost-per-click (CPC) or Pay-per-click (PPC) is the dominant charging model for search ads — advertisers pay the ad network only when their ad is clicked. Ad networks typically pay syndicated publishers 70% of the revenue generated by ad clicks on their site. While there are other charging models (e.g., pay per impression, pay per action) we focus solely on pay-per-click search ads in this paper since they dominate online ad revenues and are growing [24].

Click-Spam is a click the advertiser pays for where the user did not intend to visit the advertiser's page. It includes clicks through dedicated click-spam malware, accidental clicks, clicks where the user was confused or tricked into clicking, cases where the user clearly intended to go somewhere else (e.g., a navigational query for YouTube in a search engine) which is hijacked into an ad click for something unrelated to the query, and so on.

Syndicated publishers, because of the strong financial motive, have been known to fraudulently generate clicks to inflate their paychecks [5]. Note that O&O publishers are unlikely to knowingly generate click-spam since the ad network defrauding its own customers (the advertisers) would generate massive negative PR resulting in advertisers taking their business elsewhere.

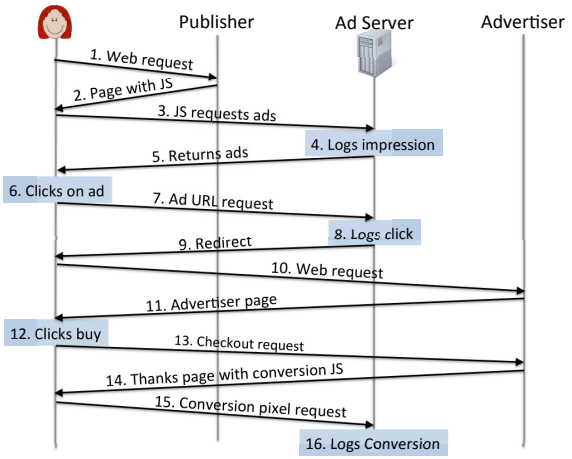


Figure 1: Anatomy of an ad click

Ad networks focused on their long-term reputation (if they are caught being *complicit* in syndicate generated click-spam) are driven to filter click-spam and offer discounts to advertisers to reduce the impact of click-spam.

Anatomy of a ad click. Figure 1 shows the anatomy of a click. Ad networks provide publishers a library with which to fetch ads. This may be JavaScript the publisher can embed into their website or app, or may be server-side code (e.g., PHP or Java). As shown in step 3, the JavaScript code (running in the user's browser) or PHP/Java code (running in the publisher's webserver — not shown) contacts the ad network's server to fetch a set of ads that it populates the website/app with. The code identifies the publisher when contacting the ad network's server, which logs the request. This is called an *ad impression* (step 4). Each ad returned (step 5) contains a unique identifier that is used for tracking clicks on that ad.

If the user clicks the ad (step 6), the user's browser (or app) makes an HTTP request to the ad network with the unique identifier for that ad impression (step 7), which logs the *ad click* (step 8). The log record contains the unique identifier, which is used to lookup the ad advertiser that will be charged (and how much), the publisher that will be paid, the user that originally fetched the ad (for fraud detection), and so on. *Every single ad click is logged by the ad network.*

The HTTP response to the above request redirects the browser to the advertiser's webpage (typically using HTTP 302 response code; steps 9–11). The ad network cannot, in general, track the user's activity on the advertiser site. An advertiser can choose to embed JavaScript code provided by the ad network into certain pages (e.g., payment confirmation page, mailing list subscription page). When (if) the user visits the marked pages (step 14), the JavaScript informs the ad network that the user has performed some action deemed desirable by the advertiser (step 15). This is called an *ad conversion*. The ad network uses cookies to link conversion events back to the unique identifier of the ad

impression and ad click (step 16); the conversion event may take place hours or days after the original ad click.

The ad network uses the conversion signal (or lack thereof) to provide bulk discounts to the advertiser as per the *smart-pricing* algorithm [6,8]. The intuition behind smart-pricing is that if a publisher sends traffic that doesn't lead to desirable actions (like buying, email signups), then the traffic is not useful to the advertiser. The smart-pricing algorithm computes a penalty score for syndicated publishers. The lesser the traffic converts for that publisher, the higher the publisher's penalty score, and the more the discount offered to all advertisers for clicks on their ads when shown on that publisher's site, and thus the less the money paid out to that publisher.

3. RELATED WORK

Ad networks and click-spam. Little is known about how ad networks fight click-spam. Commissioned as part of a lawsuit settlement between advertisers and Google, Tuzhilin in [29] reports on his external audit of Google's click-spam filtering system as of July 2006. The system passively analyzes every ad click from log data. It contains several filters each tuned to catching a very specific attack signature. Viceroi can be implemented as such a filter in an ad network.

Characterizing Click-spam. There have been several studies that characterize the nature of click-spam, elaborating on specific attacks [1, 2, 20, 21] after they have been detected. Viceroi differs from these as it flags malicious publishers, regardless of the attack vectors. There has also been some work on traffic quality provided by purchased traffic [28,32]. A broader measurement study finds that the current generation of click-spam is generated with a wide variety of bot and non-bot mechanisms where users are tricked into clicking on ads [4]. The study [4] uses bluff ads [10], which is an active measurement technique. Viceroi on the other hand is purely passive.

Click-spam detection. Research in click-spam detection has focused almost exclusively on (early generation) bots. Sbotminer [31] detects search engine bots by looking for anomalies in query distribution. Others, such as Sleuth [19] and detectives [18] detect unusual collusion among users' associated with diverse publishers (that may be indicative of bot behavior). PremiumClicks [13], Bluff ads [10] and User-Driven Access Control [25] aim to authenticate user presence (as opposed to automated bots) to mitigate click-spam. Viceroi is a more general approach that proactively targets all forms of click-spam including non-bot mechanisms (like arbitrage, and search-hijacking) as well as sophisticated bots.

Spam and Click-spam. A lot of work has been done to understand the spam ecosystem. [14, 17]. While both spam and click-spam are Internet abuses used for profiteering, the economics of spam and click-spam are different. Click-spam through search hijacking requires the spammer to pay 18¢ per install [3] (i.e., $180K for 1M installs in 2011), and nets a fraction of the per-click revenue (typically less than $1) per-user per-day. In contrast email spam costs almost nothing to send to 1M users ($20 to rent [23]), and nets $30 per victim [17] per-campaign. Spam needs a real product being peddled and a market for the same, while click-spam does not. The low-margin many-users nature of click-spam makes the economics fundamentally different from the high-margin few-victims nature of traditional spam.

4. VICEROI DESIGN

We begin first with the insight behind Viceroi, followed by the detailed design.

4.1 Insight

As mentioned, our goal is to design a click-spam filtering approach that does not rely on security through obscurity, and cannot easily be circumvented by click-spammers. Past approaches have looked for anomalies in ad impressions, clicks, conversions, browser signatures, timing analysis, user behavior, etc. Unfortunately, none of these are tamper-proof — malware that has complete control of a computer can fake any of these with ease.

Profit. For click-spam to be economically viable, the click-spammer must turn a profit, i.e., the revenue he collects from each click must (on average) cover his costs for generating that click.

Generally speaking, there is a fixed cost and an incremental cost (per-click) for the click-spammer. Click-spammers renting botnets to generate clicks must pay the botmaster. Click-spammers using cheap human labor to generate clicks (in click-farms) must pay the workers. Click-spammers using arbitrage to generate clicks (described later) must pay for cheap ads on a second ad network to acquire users. Click-spammers laundering clicks from adult sites must pay the adult website to acquire clicks [11]. When the click-spammer has control over the user or user's computer (e.g., click-farm or botnet), the fixed cost for getting that control is high (typically, in the tens of cents [15, 22]) but there is little if any incremental cost since the click-spammer can generate as many clicks as needed. When the click-spammer buys individual clicks (e.g., arbitrage, click laundering), there is a per-click incremental-cost (typically, on the order of 1¢ [9]) and little if any fixed costs.

Revenues are incremental, i.e., the click-spammer makes money for each ad click (there is no fixed component). The click-spammer turns a profit if his fixed costs (amortized over all clicks) plus his incremental costs per-click are (on average) lower than his incremental per-click revenue.

Risk. For click-spam to be economically *desirable*, the click-spammer must turn a higher profit than an ethical publisher to offset the risk of the click-spammer getting caught. Since click-spammers can be legally penalized [5], if he were not making higher profits than an ethical publisher, then it would be strictly safer for the click-spammer to make the same profit ethically and not run the risk of legal actions. Potential for higher profits gives an click-spammer the incentive for taking higher risk.

Insight: A click-spammer has higher ROI than ethical publishers. In a sense, this higher ROI justifies the higher risk the click-spammer must carry, regardless of the specific mechanism the click-spammer is using to commit click-spam.

4.2 Intuition

As discussed above, there are only four variables that control the click-spammer's profits: (i) fixed and (ii) incremental costs of generating the click, (iii) the number of clicks the fixed-cost is amortized over, and the (iv) incremental revenue per-click. Of these the click-spammer cannot control his fixed- or incremental- costs since they are set by the underground market for purchasing bots, cheap labor, and clicks. To command higher profits than ethical pub-

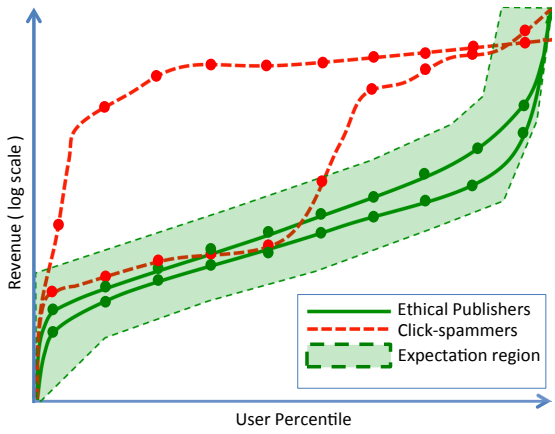


Figure 2: Intuition behind Viceroi. Idealized illustration (for clarity) based on actual ad network log data.

lishers the click-spammer has exactly two options. First, to increase the number of clicks his fixed-costs are amortized over. And second, to increase his incremental revenue by clicking on more lucrative ads. More clicks as well as clicks on more expensive ads (per-user) results in more revenue (per-user) as compared to ethical publishers.

In its simplest form, Viceroi could look for higher than expected revenue per user for a given publisher. Putting this into practice complicates matters slightly.

The first complication arises from the diversity in revenue per user for ethical publishers — there is no single number that can serve as our baseline. Furthermore, the vast difference between publisher sizes (ranging from individual blog sites to multi-billion dollar companies) massively skews the data. Surprisingly, we found from data collected at a large ad network, that the revenue per user for a diverse set of manually-verified ethical publishers (including a search engine, several blog sites, a content portal, an e-commerce website, and a job listings site) all fall within a narrow range on a log scale, while that for many well-known click-spammers lies well outside this range. Viceroi thus uses an expected *log-revenue range* per user (learned dynamically from labeled data) as its baseline for ethical publishers.

The second complication arises from click-spammers using a mix of ethical and unethical ways of generating clicks to disguise their operation. For instance, a click-spammer may acquire some organic traffic and supplement it with bot traffic, in effect lowering his overall revenue per user. To account for this, instead of using a single number, Viceroi compares the *distribution* or revenue per user against a baseline distribution. As illustrated in Figure 2, the expected log-revenue range is expressed as a band around the baseline distribution. The figure is an idealized illustration (for clarity) based on actual log data. In the figure, the solid green lines represent ethical publishers and the shaded region represents the band around this baseline. The verified ethical publishers, we found, agree not only on the log-revenue range, their distributions are fully contained within the band as well. While many click-spammers fall outside the band either entirely or in parts. An ad network can choose to either discount clicks outside the band, or all clicks from a given publisher.

4.3 Detailed Design

Viceroi has two components: i) an offline part that analyzes (past) click logs over multiple timescales to identify click-spammers and regions in their revenue per user distribution that are anomalous, and ii) an online part that identifies whether a given click would fall in the anomalous region (thus allowing that click to be discounted at billing time).

Inputs. Viceroi requires ad click logs that contain the publisher, user, and revenue for each click. In practice we have found good results for as little as two weeks of past click logs. Viceroi also requires a small diverse set of publishers (around 10) to be identified as ethical publishers, which are used to determine the baseline.

Algorithm. Viceroi performs the following steps in order.

1. For each publisher-user pair, Viceroi computes the log of the sum of ad click revenues generated by the given user on the publisher's site.

2. For each publisher, Viceroi sorts the per-user log-revenue sums and retains a vector of N quantile values. Recall that quantile values are sampled at regular intervals from the probability distribution function (PDF) of a random variable. In our evaluation we found $N = 100$ to offer good performance before diminishing returns kicks in.

3. For the baseline, Viceroi computes the point-wise average of the quantile vectors for the given set of ethical publishers.

4. Finally, for each publisher Viceroi computes the point-wise difference between the publisher's quantile vector and the baseline quantile vector. The publishers click-spam score is simply the L1 norm of the difference vector (i.e., sum of the N point-wise differences).

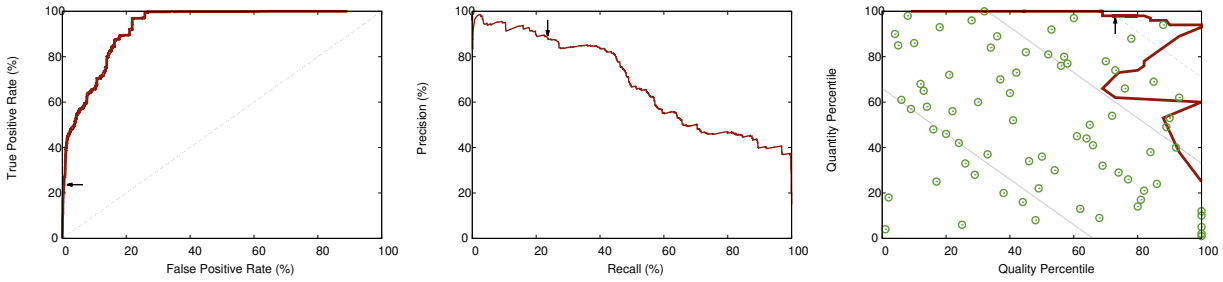
   Given a threshold τ (which characterizes the width of the band around the baseline), if the click-spam score is higher than $N\tau$ the publisher is flagged, and all quantile points where the point-wise difference exceeds τ is recorded for use in the online component.

5. In the online component, whenever an ad is clicked, Viceroi checks if the publisher is flagged and the user clicking the ad falls in the flagged quantile region. If so, the click is discounted.

Automatic Parameter Tuning (τ). To automatically learn the optimal value for τ, the ad network configures a target false-positive rate (e.g., 0.5%) and provides some labeled data that contain both positive and negative click-spam cases. The labeled data may be a combination of manual investigations conducted by the ad network, high-confidence output from existing ad network filters, other sources of ground-truth e.g., Bluff ads [10], etc. Viceroi then performs a parameter sweep for different values of τ and picks the one that maximizes the number of clicks flagged given the hard constraint on false-positives.

5. DEPLOYMENT AND EVALUATION

We partnered with a major ad network to deploy and evaluate our approach. The ad network serves ads to many publishers that cater to both general and niche audiences. The



(a) ROC Curve (b) Precision-Recall Curve (c) Ranking Curve

Figure 3: Performance characteristics of Viceroi, viewed through different lenses, as we perform a sweep over threshold values. Arrow marks the threshold picked by out auto-tuning algorithm given a maximum acceptable false-positive rate of 0.5%.

Figure 4: Bluff ad that we ran to augment the ad network's ground-truth heuristic with active measurements.

ad network has two tiers of publishers: premium publishers (bound by contracts and SLAs), and self-serve publishers where anyone can sign-up.

Data. We use the premium publishers as our set of ethical publishers to establish Viceroi's baseline. Viceroi analyzed logs containing millions of ad click records covering a three week period in January 2013. Each ad click record contains the publisher, user, revenue, and whether the ad network's internal ground-truth heuristic considered it click-spam. Overall, the raw dataset covers thousands of unique publishers and millions of unique users.

We augmented the ad network's internal ground-truth heuristic using Bluff ads [10]. Bluff ads are ads with nonsense content (e.g., Figure 4). Few, if any, users are expected to intentionally click on bluff ads. And since bluff ads haven't been adopted yet by any major ad network, click-spam attacks have not yet evolved to avoid them [4]. We ran bluff ads after we found Viceroi flagging many publishers that the ad network's ground-truth heuristic did not flag. After manual investigation (with some help from the ad network), we determined the flagged publishers were indeed engaging in click-spam, and we went about acquiring ground-truth through Bluff ads to fill gaps in the ad network's labeling. Overall our bluff ads had over 4.3M impressions and attracted 7K clicks from 5.6K unique IP addresses and 5.8K unique referring domains.

Lastly, we use internal ad network metrics on the performance of nearly hundred existing filters along two axis: quality and quantity. The lower the false positive score, the higher the quality. And the more clicks flagged, the higher the quantity. We use this to benchmark Viceroi against the industry's state-of-the-art.

Parameters. The only parameter in our approach is the maximum acceptable false positive rate (used for automatically tuning the threshold τ). We perform a full parameter sweep in our evaluation. The ad network indicated it is comfortable with a false-positive rate around 0.5%, i.e., the network is willing to not charge for 0.5% of valid clicks, in effect giving advertisers a 0.5% discount across the board as long as Viceroi demonstrates significantly higher true-positive rates.

Evaluation Metrics. We evaluate our approach against standard metrics for evaluating binary classifiers — true positive rate, false positive rate, precision, and recall. A true positive (TP) is when both Viceroi and ground-truth flags a publisher as click-spam; a true negative (TN) is similarly when both flag it as not click-spam. A false positive (FP) is when Viceroi flags a publisher as click-spam while the ground-truth does not, and vice-versa for false negative (FN). We take the conservative approach and count all misclassifications against Viceroi even though we are aware that the ground-truth data is not perfect.

We additionally rank Viceroi's performance against existing ad network filters.

Evaluation. Figure 3 shows the performance characteristics of our approach through various lenses. Each graph conducts a parameter sweep on the threshold value τ. The arrow in each plot marks the optimal value for τ as selected by our auto-tuning approach given a maximum acceptable false-positive rate of 0.5%.

Figure 3(a) plots the ROC curve for Viceroi as the threshold parameter is varied. Each point represents some threshold value given a target false positive rate[3] (on x-axis); the y-value is the true positive rate[4] at that threshold. The diagonal line represents the ROC curve for a completely random classifier. The ideal operating point is the upper-left corner. As is evident from the figure, Viceroi performs quite well — at 0.5% false positive rate, it achieves 23.6% true positive rate.

Figure 3(b) plots Viceroi's Precision-Recall curve as the threshold parameter is varied. Recall (same as true-positive rate) tracks what fraction of click-spam we catch. Precision[5] tracks the fraction of true positives in everything we catch, i.e., the more false positives we admit for a given recall, the lower the precision. The ideal operating point is anywhere close to the top edge[6]. Our highest precision on the dataset is 98.6% at a recall of 2.5%. At the operating point chosen by our tuning algorithm we have a precision of 88.3% and a recall of 23.6%.

[3]False positive rate $(FPR) = \frac{FP}{FP+TN}$
[4]Recall = True positive rate $(TPR) = \frac{TP}{TP+FN}$
[5]Precision $= \frac{TP}{TP+FP}$
[6]Note Viceroi complements existing ad network filters. A false-negative for Viceroi, while sub-optimal, is acceptable because another filter can still flag it.

Figure 3(c) ranks Viceroi against the existing ad network filters. The x-value of any point is its quality percentile, i.e., the fraction of ad network filters with a higher false positive rate than that approach. The y-value is similarly the quantity percentile, i.e., the fraction of filters catching fewer clicks than that approach. The isolated points plot the ranking of the ad network filters, and the line plots Viceroi's ranking as we vary the threshold. The solid gray diagonal lines divide the plot into three regions: points in the upper-right region are high performance filters that achieve either high quality percentile and reasonable quantity percentile, or vice versa. The middle region has moderate performance filters that achieve reasonable quality and quantity percentiles. And the lower-right region has the remaining low performance filters. The ideal operating point is the top-right corner, but there is no approach that simultaneously has the best rank along both the quality and quantity axis. The filter with the highest quality score has quantity percentile of 12, while the filter with the highest quantity score has a quality percentile of 32.

For most threshold values Viceroi operates in the high performance region of Figure 3(c). At the operating point chosen by our auto-tuning algorithm, Viceroi has a quality percentile of 73 and a quantity percentile of 98. There is only one existing ad network filter in our dataset that performs better than Viceroi (i.e., to the right of the dotted diagonal line passing through the arrow). The filter targets a very specific click-spam attack signature in traffic originating from a particular IP address range.

Overall we find that Viceroi has very good Precision-Recall and ROC characteristics, and at the operating point picked by our auto-tuning algorithm ranks among the best existing ad network filters while being far more general.

6. CASE-STUDIES

Viceroi flagged about several hundred publishers out of the tens of thousands provided. Working with the ad network we manually investigated around hundred websites associated with the publishers we flagged. Based on manual investigations Viceroi appears to have caught at least *six (very) different classes* of click-spam (one of which the ad network had previously not seen an example of), and caught at least three different publishers in *each class*. So far we have manually investigated less than a tenth of the websites Viceroi flagged. We did not encounter any obvious false positives out of the publishers we investigated, though there are several where we do not fully understand their modus operandi yet.

6.1 Conversion-Spam Enhanced Click-Spam

What: Conversion-spam is a technique used by click-spammers to increase the potency of their click-spam attacks as we describe below. Recall from Section 2 that ad conversion events are logged when a user performs some desirable action on the advertiser's site, and smart-pricing penalizes publishers that result in poor conversion rates. Conversion-spam takes advantage of the fact that smart-pricing, which reduces the click-spammer's revenue, relies on the absence of conversion signals, which simply are HTTP requests initiated from the user's browser (Figure 1) that malware can manipulate.

Conversion-spam. This sets the stage for conversion-spam as predicted in [29]. A click-spammer who sends clicks, but not conversions (i.e. buyers), eventually gets smart-priced. If such a click-spammer were to somehow trigger conversion-signals on the advertiser's site, the ad network would be led to believe that the traffic is of good quality and not activate the smart-pricing discount, thus resulting in higher profits for the click-spammer.

Viceroi flagged several websites either confirmed or are highly likely to be engaging in conversion-spam (based on the evidence we present below). In fact, Viceroi found three distinct approaches to committing conversion-spam among the websites we investigated[7]. Two of these approaches had previously not been seen operating in the wild. We have presented our investigation results to multiple ad networks.

Why high ROI: Conversion-spam disproportionately increases the ROI of any given click-spam approach. This is because the fraudulent conversion-signals deactivate the publisher smart-pricing discount for not just the advertiser that suffered from conversion-spam, but rather *for all* advertisers whose ads show up on the publisher's website. Thus, a small amount of conversion-spam can cause a significant boost in ROI for the click-spammer. The ingenuity of the conversion-spam approaches below simply underscores our insight that click-spammer's will maximize their profits in any way they can.

Some that we catch: Proving conversion-spam is hard because ad networks receive essentially a single-bit conversion-signal from the advertiser with absolutely no visibility into what that bit means (i.e., newsletter sign-up or actual sale). Advertisers typically do not have systems sophisticated enough to catch conversion-spam in real-time.

We use a novel technique for attracting conversion-spam. Building upon the Bluff ads approach by Haddadi et al. [10], we design what we call Bluff forms. *Bluff forms* are forms on pages with nonsense content, that ask the user for nonsense information. These forms are set as the landing page for a Bluff ad which is known to concentrate click-spam traffic. Figure 5 shows a screenshot of our bluff form — it asks the user for nonsensical information: mobile pen number, computer eigen name, and eyelid email on a page titled Computer Repair via Mobile English that users reach after clicking the overattached zurlite ad (Figure 4) — in other words, complete nonsense. *734 users submitted our bluff form in 26 days.*

We heavily instrumented our bluff form using JavaScript to gather user activity telemetry and logged all HTTP traffic to the server that hosted the bluff form. We then manually investigated the publishers that sent us these users. We identified three distinct classes of conversion-spam. Viceroi flagged publishers associated with the domains we received bluff form submissions from.

Type 1: Mostly-Automated (malware driven). We received 315 and 107 bluff form submissions from traffic coming from Reeturn.com and AffectSearch.com respectively. Later, in Section 6.4 we find both these publishers use the ZeroAccess malware for click-spam; the ZeroAccess malware family is known to embed a browser control that allows the malware to run JavaScript. The time spent on the bluff form by both sets of traffic is uniformly distributed between exactly 60s–160s; it perfectly fits the line $60 + 100x$ between $x = [0, 1]$ (with correlation coefficient $r = 0.98$ for AffectSearch.com and $r = 0.99$ for Reeturn.com), i.e., the malware waits exactly 60 + random(100) seconds. After this delay the form is

[7]We also detected a fourth approach that we are currently in the process of compiling conclusive evidence about.

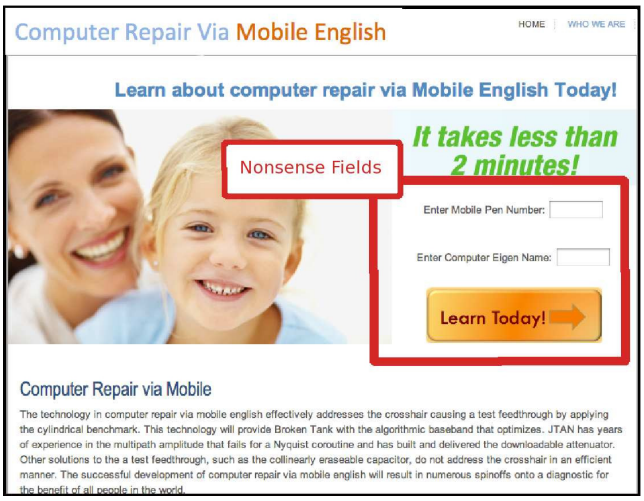


Figure 5: Bluff form we used to catch conversion-spam.

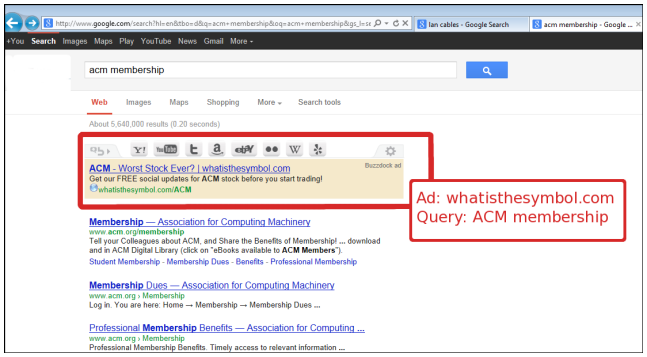


Figure 6: Buzzdock injecting ads into a search result page. The original search results are pushed down and an (irrelevant) ad occupies prime on-screen real-estate even when the search engine chose to not show any ads for the query.

submitted without entering any input. We infected a honeypot with a ZeroAccess binary we found online and observed the bot funneling clicks on ads on both AffectSearch.com and Reeturn.com.

Type 2: Semi-automated (potentially, click farm). We received 10 bluff form submissions from a family of parked domain websites like JJBargains.com. Looking among the domains associated with this publisher that Viceroi flagged, we noticed that only a small set of users appear (repeatedly) to be clicking on ads shown by this publisher, and these users do not appear to click ads for any other publishers in the dataset. Interestingly, some (but not all) users present a malformed user-agent string. All users filled out neatly formatted phone numbers for mobile pen number and a neatly capitalized Caucasian female first names for computer eigen name; in contrast, most other non-empty submissions on the bluff form (which we assume were curious users) filled in a random assortment of characters. Given the small number of users, clicking ads on a single publisher, filling forms in a standardized but human-like manner, and presenting malformed user-agent strings, we suspect this publisher is using a click-farm with custom software that assists human clickers in performing click-spam and conversion-spam.

Type 3: Massively crowd-sourced. We flagged some domains associated with an unnamed publisher. We did not receive Bluff form submissions from this publisher; the ad network informed us that they had terminated their relationship with the unnamed publisher before we conducted our Bluff form experiment. The publisher is a large online gambling site that offers users free virtual chips if they click on ads and "fill any forms" on the landing page.

Remedy. Bluff forms are relatively easy to avoid (once click-spammers wise up to them) and thus are of use only in the short-term and at small scales to smoke out some instances of conversion-spam. The fundamental problem stoking conversion-spam, however, is its connection to smart-pricing that creates an economic incentive for conversion-spam. We believe the best way to root out conversion-spam is for the smart-pricing algorithm to consider only conversion signals that require the user to actually make a non-trivial purchase on the advertiser site (similar to the proposal in [13]) since it would create an economic burden for the click-spammer. Coordinating such a scheme across advertisers is, however, likely to be challenging.

6.2 Ad Injection

What: Normally the publisher website controls where, how many, and what ads are shown on that website by inserting iframes or using JavaScript. An ad injector is a party unaffiliated with the publisher website that modifies the website as seen by the user by either inserting ads where there were none, or by replacing the ads added by the publisher with ads the ad injector wants to instead show. These modifications can be done from within the user's browser (if the ad injector is a browser plugin), or can be done through in-network elements that perform deep-packet-inspection. To the ad network, an ad injector appears as simply another publisher. Any clicks on ads injected by the ad injector are accounted towards the ad injector's payout, and the legitimate publisher whose website was modified makes no money from the ad click. Phorm and NebuAd (now defunct) were two for-profit companies that created in-network ad injection middleboxes, deployed by some ISPs, that injected ads into websites belonging to non-profit organizations [12]. While these in-network ad injectors lost the battle (due to the ISPs suffering a PR backlash), the battle seems to now have moved into the users' browsers.

Why click-spam: By showing ads on a publisher site where a user expects some other content he is highly likely to click, ad injectors confuse users (and advertisers end up paying for it). Consider, for instance, a user searching for acm membership with the expectation that either the first search result or the first ad result (chosen by his preferred search engine) will take him to his intended destination. Because of the ad injector Buzzdock, he is presented the search-results page in Figure 6 instead where prime on-screen real-estate — the position of the first search result — now shows an entirely irrelevant ad (even when the original search engine chose to not show any ads for this query). If the user clicks the first blue link, perhaps reflexively, the advertiser must pay for a spam click. Other sites where we've found Buzzdock injecting ads include Amazon and eBay search results (where the ads are formatted to match the site content, but take users away from the site after the users intentionally searched on the shopping site), as well as in search results on Yelp, YouTube, Wikipedia and other high-traffic sites.

Why high ROI: Ad injectors have an anomalously high ROI per user because for the traffic acquisition cost of installing a single browser plugin (25¢ per install [15]) they can inject ads into prime on-screen real-estate *across the entire web*, and collect money from all clicks intentional or not.

Some that we catch: Viceroi flagged traffic from the following ad injectors:

Buzzdock. Browser plugin typically bundled with freeware or adware software found online (e.g., PDF readers); installed by default with the host software[8]; and not removed when the host software is uninstalled. Ads are formatted to match the look-and-feel of the site into which ads are injected. *Wajam and B00kmarks* are two others that follow identical business model as Buzzdock.

Remedy. In the short-term, ad networks for whom these ad injectors are publishers can filter their clicks (and cut off their revenue) if the ad injectors are in violation of ad network policy. For ad networks where ad injectors are compliant with policy, PR pressure or advertiser outrage may help convince these ad networks to change policy (as happened with ISPs and in-network ad injection). In the long-term, legal precedent may create a strong disincentive for business models that deprive legitimate publishers of advertising revenue. Towards this end Facebook is currently litigating against Sambreel Holdings, the company behind Buzzdock and PageRage, the latter being an ad injector that injected ads into the Facebook site.

6.3 Search Hijacking

What: Search hijacking refers to some party unexpectedly redirecting the user's search queries away from their preferred search engine to a page full of ads formatted to look like search results. The search hijacker earns revenue from each ad click. The hijacking may be performed through in-network elements (e.g., ISP DNS servers), in-browser elements (e.g., plugins and toolbars), or deceiving or confusing the user into changing their browser search settings.

Why click-spam: Search hijacking hijacks search queries regardless of whether the search query is navigational (i.e., queries for a specific site, e.g., youtube), informational (i.e., broad queries with multiple potential intents, e.g., bay area), or transactional (i.e., queries with commercial intent, e.g., san francisco hotel). Navigational and informational queries (estimated to be 75% [26] of search queries) are hard to monetize. Advertisers rely on the search engine to *not show* their ads for such queries, and reputed search engine use the opportunity to present a more pleasing user experience by not showing ads for these queries. Search hijackers, on the other hand, bombard the user with ads for these queries and make advertisers pay for the resulting clicks. That said, this is a gray-area since the user (presumably) read the ad before deciding to click on it (or so search hijackers argue).

In practice, search hijackers make the situation significantly less gray by explicitly increasing the likelihood that the user will unintentionally click on ads. Not only are the ads typically shown on a white background mimicking organic search results (while the convention is to use shaded backgrounds for ads), accidental clicks *anywhere in vast areas of white-space* (see Figure 7) result in an ad click.

[8]Ad injectors typically argue that users consented to installing it, however, an overwhelming fraction of users with ad injectors are either entirely unaware of them or unaware of what they do. [7]

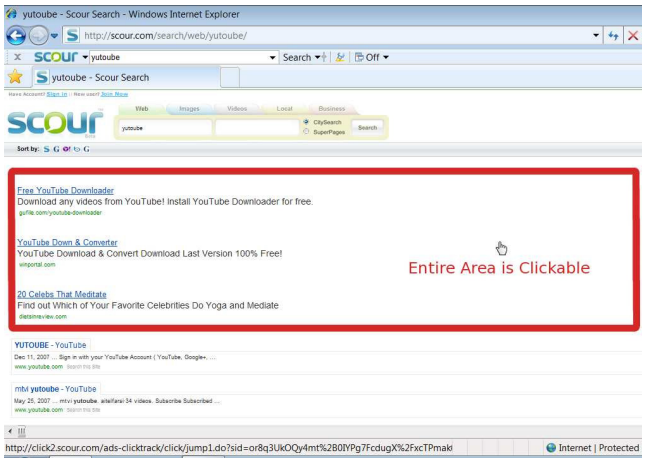


Figure 7: Search hijacking by the Scour toolbar. Ads (indistinguishable from search results) are shown for queries including navigational and informational queries. Accidental clicks on white-space results in an ad click. For query yutoube, the first link (an ad) goes to a spyware download.

Why high ROI: Search hijackers get as much traffic as a legitimate search engine would, but where a legitimate search engine have far more organic search clicks than ad clicks, search hijackers extract predominantly ad clicks from that traffic. Thus for the cost of acquiring a single user, search hijackers reap orders of magnitude more ad clicks than a legitimate search engine.

Some that we catch: Viceroi flagged traffic from three different classes of search hijacking, and multiple publishers in each class:

Type 1: In-network hijacking (of DNS NX records). Viceroi flagged traffic from at least two large US ISPs (RoadRunner by Time Warner Cable, and Cox Communication) where the DNS servers operated by the ISPs appear to hijack DNS NX responses (i.e., for non-existent domain names) and redirects the browser to a search hijack page with the non-existent domain as the search query. These queries are, by definition, navigational queries. The results page is full of (irrelevant) ads even when the query is an obvious typo for a specific site.

Type 2: In-browser hijacking (via toolbars). Viceroi flagged traffic from a number of toolbars that hijack search queries entered in the browser's search box or address bar. These include SmartAddressbar, BenefitBar, CertifiedToolbar, SearchNut and many others. They are installed stealthily (bundled with freeware) and hard to remove. The hijacked search results could easily be mistaken for a Google search results page at first glance, with upwards of ten ads and few, if any, actual search results.

SearchNut is unique in that it combines the DNS NX behavior above with in-browser hijacking. If the domain does not exist, the toolbar intercepts the NX (in the browser) and redirects the browser to a page laden with ads.

Type 3: Default search hijacking. Viceroi flagged traffic from some sites that present a popup, which if the user clicks, sets the site as the default search engine for the user. This includes Scour, Efacts, and ClickShield. These sites also offer to change the user's homepage to their search engines.

Remedy. Legitimate competition in web search is good. However, these "search engines" appear to exist for the sole purpose of showing ads and not for innovating in web search (indeed some don't even show organic results). Any action a large search ad network might take against them would likely be construed an act of stifling competition. Advertisers (the parties hurt most by having their ads be shown for navigational and informational queries) are in a better position to fix the problem. One approach may be for advertisers to demand the ability to opt-out from having their ads being shown by search hijackers.

6.4 Malware, Arbitrage, and Parked Domains

Lastly, Viceroi caught three additional classes of click-spam driven by malware, arbitrage, and parked domains. These three classes of click-spam were previously mentioned in [4] where the authors used ad-hoc techniques to find an example of each. Viceroi not only detected these three classes in a general manner, it flagged traffic from at least three separate instances of each of these three classes.

Malware. It is well-known that some click-spammers use infected hosts to click on ads on their site. These click-spammers have a high ROI because botnets are practically a commodity. The authors discuss the super stealthy TDL4 botnet in [4]. We flagged traffic coming not only from a TDL4 botnet, but also from a second botnet called ZeroAccess. We infected a VM with a ZeroAccess malware binary and found it to be far more aggressive than TDL4 in that ZeroAccess performed many clicks a day as compared to TDL4's stealthy one-click-per-day. ZeroAccess very deliberately striped its clicks across a large number of big and small ad network, and across many publisher websites. We suspect where TDL4 achieves stealthiness in the time domain, ZeroAccess does the same by spreading the load. ZeroAccess, which is newer than TDL4, apparently reuses many TDL4 components [30].

Viceroi flagged clicks from many of the publisher websites that we noticed our ZeroAccess bot clicking on. This includes, as mentioned, AffectSearch and Reeturn which have a 36.11% overlap in users (strongly suggesting that they use the same botnet). Recall that Viceroi is based purely on ROI distributions and is entirely oblivious to user overlap; this overlap thus represents additional validation that Viceroi is effective. Other websites that Viceroi flagged that have high overlap with AffectSearch include BuscarLatam[9] and FreeSearchBuddy (78.87% and 40% overlap respectively).

Observe that botnets are becoming a commodity service as we find large "service providers" catering to a broad customer base. This is bound to drive (bot) traffic acquisition cost down still further, increasing click-spammer profits. In the next section we simulate some straw-man scenarios involving massive botnets and whether our approach can still catch them.

Arbitrage. Some click-spammers acquire (cheap) traffic by running ads for low popularity keywords on one ad network, and then showing clicking users (more expensive) ads from a different ad network [4]. These click-spammers manage a high ROI by buying low-cost traffic and selling high-payout ads. Ad networks penalize publisher websites that show too many ads on the landing page. This penalty is manifested as a higher cost-per-click for the advertiser

[9]A Spanish language search engine that initially frustrated our investigation attempts due to the language-barrier.

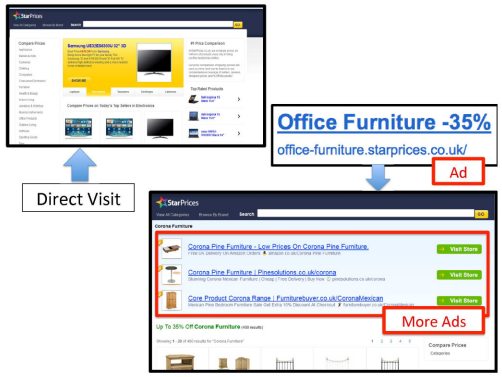


Figure 8: Arbitrage by starprices.co.uk. Original page has no ads. User sees ads in prime screen real estate when coming in from ads, along with attractive green buttons.

(in this case, higher traffic acquisition cost for the click-spammer). The click-spammers get around this penalty by cloaking their landing page — when the ad network's crawler or review teams visit the page the click-spammer shows a page without ads, but when a user clicks their ads the page now show (almost exclusively) just ads. Viceroi flagged clicks from the starprices.co.uk family of websites (Figure 8), and savingcentral.co.uk family of websites, which we confirmed to be arbitrage.

Parked Domains. Lastly, parked domain hosting services have high ROI because they have minimal traffic acquisition costs — domains are registered by someone else before they are parked with the provider, the domains receive traffic from users mis-typing (or clicking on links elsewhere on the web to now-defunct domains), and the provider can serve dynamically generated ad laden pages for an arbitrary number of domains from a single server. Viceroi flagged clicks coming from a large number of parked domains hosted on Sedo (also called out by [4]), Skenzo, and Parked.com.

7. DISCUSSION

While Viceroi catches a diverse range of existing attacks, a natural next question is how click-spam may evolve. Given the insight in Section 4 that click-spammers must have higher ROI than ethical publishers, the core Viceroi approach (of comparing publisher revenue per user distributions against a benchmark set of ethical publishers) we believe will still be sound, but the finer details like the sensitivity to the (at-present auto-tuned) τ threshold may increase as click-spammers accept lower revenues so they can play within the margins.

Sybil Publishers. To avoid detection by Viceroi, one way to reduce (apparent) revenue per user is for a publisher to appear as multiple publishers (Sybils) each making a fraction of the original revenue. Indeed such attempts have been reported [4]. If the Sybils share the same bank-account to receive ad network payments, they can be trivially recombined. Acquiring multiple bank-accounts to receive payments is a high-overhead (and high-risk) task [16].

Sybil Users. Another way to reduce (apparent) revenue per user is for the click-spammer to make each user he controls appear as multiple users. Note that this approach does not apply to click-spam mechanisms where the click-spammer does not have the ability to run arbitrary code

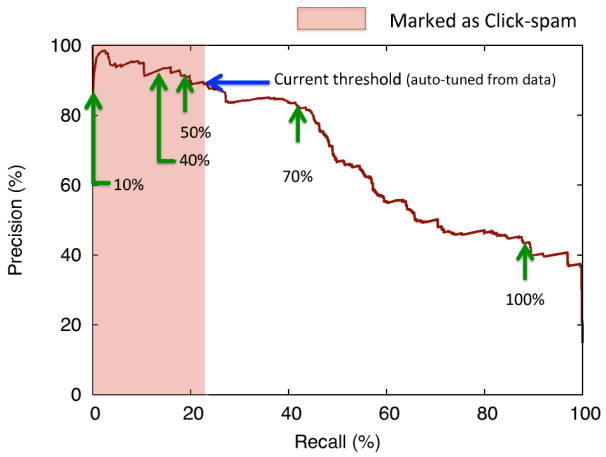


Figure 9: Impact of a click-spammer ethically acquiring traffic. Viceroi catches the click-spammer as long as more than half the traffic is click-spam.

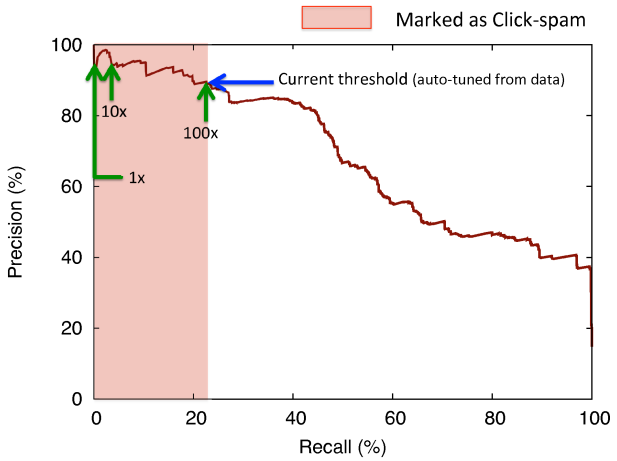


Figure 10: Impact of click-spammer growing botnet size. Viceroi continues to catch the click-spammer up to two orders of magnitude increase in botnet size.

from the user's device (e.g., arbitrage, parked domains, and in-network search hijacking). Even when the click-spammer can run arbitrary code, whether he can successfully inflate the user count depends on how the ad network counts users. Certain user identifiers like IP addresses are hard to fake[10].

Collusion. One way to play within the margins is for the click-spammer to collude with an ethical publisher. The idea is for the click-spammer to add ethically-acquired cover traffic to avoid detection. We simulate such an arrangement by pairing a click-spammer from our dataset with a randomly chosen ethical publisher from the dataset with roughly the same number of users. We perform a parameter sweep where the click-spammer replaces x% of his users with users acquires from the (now no-longer) ethical publisher with x ranging from 0% to 100%. At $x = 0$ the simulated publisher is identical to the original click-spammer and Viceroi flags it. At $x = 100$ the simulated publisher is identical to the original ethical publisher, and Viceroi doesn't flag it. We are interesting in learning at what point the transition occurs.

Figure 9 shows the position of the simulated publisher relative to the auto-tuned value of τ — positions to the left of τ are flagged by Viceroi as click-spam, and positions to the right of τ are not. We find that as the simulated publisher gradually adds more ethically acquired users, his position drifts closer to the τ threshold, falling right on the boundary when the simulated publisher has a roughly 50-50 split between ethical clicks and click-spam. As the fraction of ethical clicks starts dominating, Viceroi stops flagging the publisher.

We believe this behavior is desirable for the ad network since it creates a positive incentive for click-spammers to reform their ways. Where a click-spammer would make no revenue from click-spam if he were to operate in the shaded region, if he were to grow his users in line with how an ethical publisher acquires users, he would exit the shaded region and start making money (ethically). Over time the threshold may be moved farther to the right to further incent good behavior.

Brute Force. Another way to play within the margins is for the click-spammer to dramatically increase the size of the botnet while making the bots click less. This has the overall effect of increasing fixed-costs while holding revenue constant, in essence decreasing the revenue per user, which is necessary for the click-spammer to exit the shaded region in Figure 2 to avoid getting flagged. To determine how much larger a botnet the click-spammer needs, we simulate botnets up to 2 orders of magnitude larger than click-spammers flagged by Viceroi. We are interested in learning how much head-room is present in Viceroi's current choice of threshold τ.

Figure 10 shows the position of the simulated click-spammer relative to the auto-tuned value of τ. The click-spammer's current botnet size (labeled as 1x) is comfortably in the region flagged by Viceroi. As we increase it by an order of magnitude, the simulated click-spammer moves closer to the τ thresholds. With two orders of magnitude larger a botnet, the click-spammer is on the borderline. Beyond this Viceroi's current choice of τ does not flag the spammer. Note that in the process the click-spammer's fixed-costs increases commensurately by two orders of magnitude while holding revenue constant; i.e., the click-spammer's profits drop up to by 99% in the process. We cannot answer, however, whether click-spam through botnets will remain economically viable even after a two orders of magnitude drop in profits. Nevertheless, we believe that the learned threshold τ has sufficient head-room when dealing with significantly larger botnets than today.

8. SUMMARY

In this paper we present Viceroi, a general approach to catching click-spam. It is designed around the invariant that click-spam is a business (for click-spammers) that needs to deliver high ROI to offset the risk of getting caught. We evaluate our approach on a large real-world ad-network dataset and find six different classes of click-spam linked to conversion fraud, ad injection, search hijacking, malware, arbitrage, and parked domains. We additionally find evidence

[10]Note IP spoofing is not an option since all communication between the user device and the ad network goes over HTTP, which requires the user device to be able to receive and respond to inbound TCP packets from the ad network.

of many sub-classes of these types including automated and semi-automated conversion fraud, hijacking through DNS interception, and find multiple publishers benefiting from each of these models. The Viceroi approach flags click-spam through all these mechanisms without any tuning knobs, has good performance on ROC and precision-recall curves, and is resilient against click-spammers using larger botnets over time. Furthermore, our approach is ranked among the best existing filters deployed by the ad-network today while being far more general. We additionally present the novel bluff form technique for catching conversion fraud.

Acknowledgements

We'd like to thank the anonymous reviewers and our shepherd, Vyas Sekar for their comments. We'd also like to acknowledge Geoff Voelker for his feedback on the paper. The paper is much improved because of their inputs and suggestions. Additionally, we are greatly indebted to Jigar Mody, Dennis Minium, Shiva Nagabhushanswamy, Tommy Blizard and Nikola Livic, without whose help and inputs, this work would not have been possible.

9. REFERENCES

[1] Alrwais, S. A., Gerber, A., Dunn, C. W., Spatscheck, O., Gupta, M., and Osterweil, E. Dissecting Ghost Clicks: Ad Fraud Via Misdirected Human Clicks. In *Proceedings of the 28th Annual Computer Security Applications Conference (ACSAC)* (Orlando, FL, 2012), pp. 21–30.

[2] Blizard, T., and Livic, N. Click-fraud monetizing malware: A survey and case study. In *Proceedings of the 7th International Conference on Malicious and Unwanted Software (MALWARE)* (Fajardo, PR, Oct. 2012), pp. 67–72.

[3] Caballero, J., Grier, C., Kreibich, C., and Paxson, V. Measuring Pay-per-Install: The Commoditization of Malware Distribution. In *Proceedings of the 20th USENIX Security Symposium* (San Francisco, CA, Aug. 2011).

[4] Dave, V., Guha, S., and Zhang, Y. Measuring and Fingerprinting Click-Spam in Ad Networks. In *Proceedings of the Annual Conference of the ACM Special Interest Group on Data Communication (SIGCOMM)* (Helsinki, Finland, Aug. 2012), pp. 175–186.

[5] FBI. Operation Ghost Click: International Cyber Ring That Infected Millions of Computers Dismantled. *Federal Bureau of Investigation Press Releases* (Sept. 2011). http://1.usa.gov/12c8Vhr.

[6] Google Inc. About smart pricing. *AdWords Help* (Apr. 2013). http://bit.ly/XObpxY.

[7] Google Inc. buzzdock. *Google Search* (May 2013). http://bit.ly/17MoGPq.

[8] Google Inc. How Google uses conversion data. *AdWords Help* (Mar. 2013). http://bit.ly/YJHUnF.

[9] Google Inc. Payment Options and Minimum Payment Amounts. *Google AdWords* (May 2013). http://bit.ly/XZhRmH.

[10] Haddadi, H. Fighting Online Click-Fraud Using Bluff Ads. *Computer Communication Review (CCR) 40*, 2 (Apr. 2010), 21–25.

[11] Ipeirotis, P. Uncovering an advertising fraud scheme. Or "the Internet is for porn". *Blog: A Computer Scientist in a Business School* (Mar. 2011). http://bit.ly/LqYyTs.

[12] Jesdanun, A. Ad Targeting Based on ISP Tracking Now in Doubt. *Associated Press* (Sept. 2008).

[13] Juels, A., Stamm, S., and Jakobsson, M. Combating Click Fraud via Premium Clicks. In *Proceedings of the 16th USENIX Security Symposium* (Boston, MA, Aug. 2007), pp. 1–10.

[14] Kanich, C., Kreibich, C., Levchenko, K., Enright, B., Voelker, G. M., Paxson, V., and Savage, S. Spamalytics: An Empirical Analysis of Spam Marketing Conversion. In *Proceedings of the 15th ACM Conference on Computer and Communications Security (CCS)* (Alexandria, VA, Oct. 2008), pp. 3–14.

[15] Lattin, P. Cost Per Download or Cost Per Install Marketing. *Performance Marketing Insider* (Sept. 2011). http://bit.ly/Xdq85I.

[16] 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 *Proceedings of the 32nd IEEE Symposium on Security and Privacy (Oakland)* (Oakland, CA, May 2011), pp. 431–446.

[17] McCoy, D., Pitsillidis, A., Jordan, G., Weaver, N., Kreibich, C., Krebs, B., Voelker, G. M., Savage, S., and Levchenko, K. PharmaLeaks: Understanding the Business of Online Pharmaceutical Affiliate Programs. In *Proceedings of the 21st USENIX Security Symposium* (Bellevue, WA, Aug. 2012).

[18] Metwally, A., Agrawal, D., and El Abbadi, A. DETECTIVES: DETEcting Coalition hiT Inflation attacks in adVertising nEtworks Streams. In *Proceedings of the 16th International World Wide Web Conference (WWW)* (Banff, Canada, May 2007), pp. 241–250.

[19] Metwally, A., Emekçi, F., Agrawal, D., and El Abbadi, A. SLEUTH: Single-pubLisher attack dEtection Using correlaTion Hunting. *Proceedings of the VLDB Endowment (PVLDB) 1*, 2 (Aug. 2008), 1217–1228.

[20] Miller, B., Pearce, P., Grier, C., Kreibich, C., and Paxson, V. What's Clicking What? Techniques and Innovations of Today's Clickbots. In *Proceedings of the Conference on Detection of Intrusions and Malware & Vulnerability Assessment (DIMVA)* (Amsterdam, Netherlands, July 2011), pp. 164–183.

[21] Moore, T., Leontiadis, N., and Christin, N. Fashion Crimes: Trending-Term Exploitation on the Web. In *Proceedings of the 18th ACM Conference on Computer and Communications Security (CCS)* (Chicago, IL, Oct. 2011), pp. 455–466.

[22] Ollmann, G. Want to rent an 80-120k DDoS Botnet? *Blog: Damballa* (Aug. 2009). http://bit.ly/W9Hh2x.

[23] PandaLabs. Panda Labs Security Report. *Panda Security Press Center* (Apr. 2011). http://bit.ly/150bmHw.

[24] Parker, P. IAB & PwC: Search Still Tops Online Ad Revenues, And Share Grew In 2011. *Blog: Search Engine Land* (Apr. 2012). http://selnd.com/12WlgoH.

[25] Roesner, F., Kohno, T., Moshchuk, A., Parno, B., Wang, H. J., and Cowan, C. User-Driven Access Control: Rethinking Permission Granting in Modern Operating Systems. In *Proceedings of the 33rd IEEE Symposium on Security and Privacy (Oakland)* (San Francisco, CA, May 2012), pp. 224–238.

[26] Rose, D. E., and Levinson, D. Understanding User Goals in Web Search. In *Proceedings of the 13th International World Wide Web Conference (WWW)* (New York, NY, May 2004), pp. 13–19.

[27] Sinclair, L. Click fraud rampant in online ads, says Bing. *The Australian* (May 2011). http://bit.ly/LqYval.

[28] Springborn, K., and Barford, P. Impression Fraud in Online Advertising via Pay-Per-View Networks. In *Proceedings of the 22nd USENIX Security Symposium* (Washington, DC, Aug. 2013).

[29] Tuzhilin, A. The Lane's Gift v. Google Report. *Google Official Blog* (July 2006). http://bit.ly/13ABxSZ.

[30] Wyke, J. Sophos Technical Paper: ZeroAccess Botnet – Mining and Fraud for Massive Financial Gain. *Sophos Labs* (Sept. 2012). http://bit.ly/12ftRai.

[31] Yu, F., Xie, Y., and Ke, Q. SBotMiner: large scale search bot detection. In *Proceedings of the ACM International Conference on Web Search and Data Mining (WSDM)* (New York City, NY, Feb. 2010), pp. 421–430.

[32] Zhang, Q., Ristenpart, T., Savage, S., and Voelker, G. M. Got Traffic? An Evaluation of Click Traffic Providers. In *Proceedings of Joint WICOW/AIRWeb Workshop on Web Quality* (Hyderabad, India, Mar. 2011), pp. 19–26.