Online Advertising Fraud

Neil Daswani, Chris Mysen, Vinay Rao, Stephen Weis,
Kourosh Gharachorloo, Shuman Ghosemajumder,
and the Google Ad Traffic Quality Team

From the forthcoming book, Crimeware,
edited by Markus Jakobsson and Zulfikar Ramzan,

The growth of the web-based online advertising industry has created many new opportunities for lead generation, brand awareness, and electronic commerce for advertisers. In the online marketplace, page views, form submissions, clicks, downloads, and purchases often result in money changing hands between advertisers, ad networks, and web site publishers. Since these web-based actions have financial impact, criminals have also seeked to take advantage of new opportunities to conduct fraud against these parties with the hopes of having some money illegitimately change into their own hands. This chapter discusses some of the many ways, including using crimeware [1], that fraudsters attempt to leverage to defraud various parties involved in the online advertising marketplace. We also discuss countermeasures that ad networks have put in place to mitigate such fraud.

1 History

We first provide, in this section, an abridged history of online advertising, as a precursor to discussing online advertising fraud in later sections. This section is not meant to be comprehensive, but does provide some high-level background as to how online advertising via search engines emerged.

Since the commercialization of the world-wide web (WWW) in the mid-1990's, the marketplace has explored many alternatives to monetize online content and generate sales leads. During the early stages of commercialization, many companies made online brochures about their products and services available. Some early web pages provided email addresses and phone numbers as contact information, and others allowed users to fill out online forms to gather information about customer's needs and contact information as a means of lead generation. These interactive product brochures were an early form of online advertising and still are very prevalent among low-end web sites.

Companies also had to find robust ways of attracting users to their web sites to begin with. In addition to adding URLs to offline information provided to customers, companies would often add ".com" to their names during the Internet boom that occurred from 1996 to 2001 to advertise their "destination" web sites. To help users find information, products, and services to help satisfy their needs, several directories (e.g., the Yahoo! directory) and keyword search engines (e.g., AltaVista) were developed to help direct users to destination sites. Companies could request or pay to be listed in online directories, and could optimize their web sites to show up prominently in search results provided by keyword search engines.

Directories sought to monetize their services by displaying banner ads linked to destination pages. Companies that offered such advertising opportunities boasted that advertisements could be "targeted" more so than in other forms of advertising. Such companies attempted to do targeting based on user behavior,

[1] For example, clickbots, as described in Section 4.2 are an example of crimeware that can be used to conduct online advertising fraud.

and attempted to classify users into various categories. However, the targeting was not very successful because making inferences about which categories visitors to a web site should be placed into was hard based on their browsing behavior alone.

On the other hand, search services provide much easier methods of targeting; when a user enters a keyword search term, the user is directly telling a search engine what she is interested in, and the need to categorize users based on browsing behavior becomes less necessary for successful targeting. In addition, keyword search engines that displayed textual ads (as opposed to banner ads) would not only provide a user experience that was more targeted, but was also less intrusive to users. The combination of better targeting and better user experience resulted in successful online advertising offerings via search engines.

2 Revenue Models

The display and accounting of ad revenue from an online advertising campaign are generally done in one of three major categories:

- *Cost per mille (CPM).* The advertiser is charged per thousand impressions.
- *Cost per click (CPC).* The advertiser is charged per click.
- *Cost per action (CPA).* The advertiser is charged per (predetermined) action (e.g., an online sale).

In this section, we describe each of the three models. We also discuss how *syndication* and *referral* deals can be used to derive revenue from online advertising activities.

2.1 Impression-Based

Advertisers often pay search engines or online magazines a fixed price for every 1000 banner ads that are displayed, termed as *CPM*, or *cost per mille*. When an ad is displayed, an *ad impression*, or simply an *impression* is said to have taken place. The term impression is also used to refer to the ad creative itself that is displayed.

Charging and accounting by CPM is theoretically easy to implement, since each web server log entry for a banner ad image represents one impression. In practice, CPM advertising can be quite complicated, with various web servers involved along the path of delivering a banner to an end user. In addition, web page caching needs to be disabled via HTTP response headers, and HTTP proxies need to abide by such headers. Finally, measuring the effectiveness and results of a CPM-based advertising campaign can be hard due to a greater emphasis on branding campaigns and offline sales, as well as due to impression spam.

Impression spam results from HTTP requests for web pages that contain ads, but that do not necessarily correspond to a user viewing the page. For

instance, a web page crawler or "scraping" program might issue an HTTP request for a web page that happens to contain ads. Such a request would need to be distinguished from those requests issued by human users. Additionally, an advertiser may not receive any feedback when ad impressions occur and may need to completely trust the content provider or ad network's impression data.

Nevertheless, even during the early commercialization of the Internet, the marketplace immediately recognized that online advertising could improve accountability and return on investment (ROI) over traditional, offline advertisements such as on billboards or television. Each click on a banner ad indicates that an online advertising campaign is driving traffic to an advertiser's site. Online content providers and search engines provided advertisers with click-through rates (CTR), which is the ratio of users' clicks per ad impression. CTRs often depend upon the "quality" of banner ads. In practice, CTRs for the early banner ads were often low. Low CTRs could be partially attributed to poor targeting, but may also have been due to technical factors, such as longer download times for large banner ad images by users on dial-up lines.

In addition to low CTRs, another challenge faced by the online advertising industry was advertising sales. It may have not been economical for a small online magazine to hire a sales force to sell its banner ad space to advertisers. This motivated the creation of advertising networks that could amortize the cost and overhead of an advertising sales force across large numbers of small web sites. Notably, in the late '90s, DoubleClick arose as a top-tier banner advertising network. Web sites would lease web page space to DoubleClick, and DoubleClick would find and manage advertisers that paid for ad impressions and clicks on those web pages.

Ad targeting has always been an important aspect of online advertising. Higher CTRs were typically evidence that the topic of the ad was related to the topic of the web page upon which the ad was placed. The marketplace realized that advertisers would pay more not only based on placement of ads, but also for better targeting.

In the year 2000, Google launched the AdWords CPM-based advertising platform in which textual ads were targeted based upon the keywords for which a user would search. Google initially formed contracts with advertisers to display their ads on a CPM basis on the top of search results pages. In addition, Google auctioned off the ad slots on the right-hand side of search results pages, and eventually did the same for ads shown on the top of search results pages. A market for charging advertisers based not on the number of ad impressions shown, but on the number of times users clicked on ads emerged.

2.2 Click-Based

In 1998, Goto.com, a startup that arose out of the *idealabs* business incubator, developed a paid-for-placement search engine where advertisers bid to be the top-placed ads next to search results. Advertisers would pay-per-click (PPC) for their search result ads. The amount the advertiser pays per click is referred to as their cost per click (CPC). In 2001, Goto.com was renamed Overture, and

in 2003, was purchased by its largest customer, Yahoo.

In the year 2002, Google re-launched AdWords as a PPC platform. However, Google not only took advertisers' bids into account, but also took into account an ad's click through rate. In particular, advertising slots on Google's search results pages are not simply auctioned off to the highest bidder. Instead, an ad slot is allocated to the advertiser whose bid times the predicted CTR of the ad (as computed by the ad network) is highest. As a result, ad placement is not simply dependent upon how much an advertiser was willing to pay, but also on the "quality" of the ad, as determined by how often users clicked on the ad (in addition to many other factors). In effect, each ad click from a user serves as an implicit vote of the relevance of the ad to the user's query. The result is that not only do users receive more relevant ads, but Google increases revenue by better targeting advertisements.

When advertisers pay-per-click, it is important that clicks deliver value and ROI to the advertiser. Some clicks might be "invalid" or "fraudulent" and we dedicate Sections 3 and 4 of this chapter to defining and discussing sources of invalid and fraudulent clicks. Section 5 discusses high-level countermeasures to deal with "click fraud." Fraud exists in both the CPM and the CPC business models; in the former, we speak of "impression fraud" while in the latter of "click fraud."

To complement PPC advertising on its search result pages, Google in 2003 launched AdSense, an online advertising product that allowed web publishers to monetize their content by allowing Google to place ads on their web sites. To ensure the ads are relevant to the user's interest, the entire web page containing the ad slots is analyzed by AdSense to determine a set of relevant topics, and ads about those topics are inserted into the ad slots on the web page. Google pays such publishers a revenue share of the CPC paid by the advertisers when the ads are clicked. While the creation of AdSense helped publishers derive revenue for the development of their content, it also introduced additional incentives for fraudulent click activity which we will discuss in Section 3.

2.3 Action-Based

A more generic online advertising model is pay-per-action, in which the advertiser pays a cost-per-action (CPA), where "action" can be defined as the user arriving at a particular "landing" page on the advertiser's site, or the user engaging in a commercial transaction. Strictly speaking, CPC-based advertising is just a special case of CPA in which the "action" is the user clicking on an ad. However, when the term CPA is used, it typically refers to a more involved action than simply clicking on an ad, and usually connotes that an advertiser is paying based on a commercial transaction. Some suspect that CPA-based advertising might be less susceptible to click fraud, since fraudsters may need to engage in commercial transactions in order to successfully defraud advertisers. At the same time, if the commercial transaction is not very costly to induce, CPA-based advertising may be just as susceptible to click fraud as CPC-based advertising.

From an advertiser's standpoint, CPA-based advertising can be attractive because advertisers are only billed for pre-defined user actions, such as if a user makes a purchase or generates a sales lead. Advertisers only make payment to an ad network once they have derived value from a click.

However, while CPA-based advertising can be well suited for commercial sites, it has some drawbacks. Some advertisers may not have online commerce web sites. For example, they may be advertising simply for brand-name recognition to increase offline sales (e.g., typical car manufacturers). Also, an advertiser's web site might have usability issues, high latency, or simply suffer from intermittent failures which may serve as barriers to users completing transactions. Much of the risk involved in a CPA-based advertising model, then, falls on the ad network.

It is also important to consider the affect of CPA on publishers. For every impression displayed on a publisher's page, a bet is being made on which ads will generate the most revenue for that publisher. When choosing between CPM-based advertisers, such a decision is simple– chose the impression that pays the highest CPM. When incorporating CPC-based advertisers, the sustained CTR of the advertiser is easy enough to determine with a relatively small amount of data, and this helps come up with an eCPM (expected CPM) estimate. However, CPA advertising, especially for small advertisers, is difficult to translate into a reliable eCPM. As a result, small CPA advertisers are often not well suited for many publisher's sites and have a harder time getting significant traffic from their campaigns.

2.4 Syndication

Ad networks can increase their reach by syndicating the ads they deliver to other destination sites, such as search engines and other web properties. In a syndication deal, an ad network (that may be run by a search engine) provides a data feed in which the syndicator receives URLs for ad impressions. The syndicator earns a share of the CPC paid by the advertiser when the URLs are clicked on. For example, a hypothetical search engine (*hyp-search.com*) might run a PPC ad network. Another search engine *syn-search.com* that does not have an ad network of its own might enter into a syndication relationship with *hyp-search.com*. Whenever *syn-search.com* receives a query from a user, in addition to providing its search results, it sends the query to *hyp-search.com* via a data feed, and receives ad impression URLs that are relevant to the query. Then, *syn-search.com* displays the ad impression URLs on its results pages and receives a share of the CPC the advertiser pays to *hyp-search.com* when users click on the ads.

In Figure 1, we conceptually depict the interactions between the ad network, the syndicator, and the user. (To keep the conceptual depiction simple, note that the arrows in the figure do not correpsond to HTTP requests between web sites, but simply depict the flow of data and money.) Upon receiving a query from the user, *syn-search.com* relays the query onto *hyp-search.com*, and then relays the ad impression URL received from *hyp-search.com* to the user. If and when

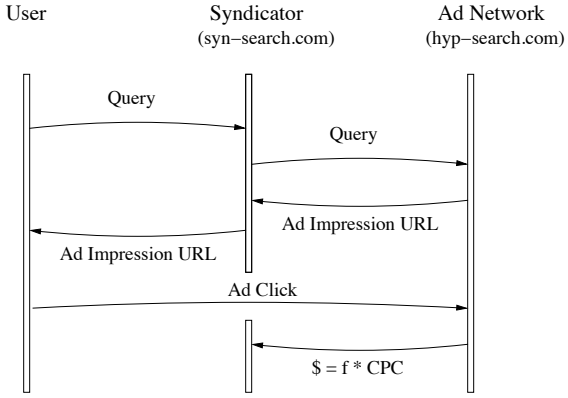

Figure 1: Syndication

the user clicks on the ad impression URL, *hyp-search.com* pays *syn-search.com* a fraction (f) of the CPC paid by the advertiser. Note that, although not shown in the figure, a click on an ad redirects the user to the advertiser's web site (i.e., the click request is sent to the ad network and generates a redirect response).

2.5 Referral Deals

In a referral deal, a web site, A, pays another web site, R, for sending web traffic to A. Web site R puts links to web site A on its web pages. In a CPC referral deal, web site A pays web site R a referral fee every time that a user clicks on a link on R's web pages and arrives at web site A. To provide a simple example, a web site called *great-online-magazine.com* might pay *ref-site.com* a referral fee every time that it displays a link to *great-online-magazine.com* on a page, and that link is clicked on by a user. Note that, in general, when a site R puts a link to A on its site as part of a referral deal, the link does not necessarily need to be an "ad," but simply can be used to encourage visitors to site R to visit site A thereafter. Reiter et. al. discuss how abuse can take place in referral deals in [RAM98].

An important distinction between CPC referral deals and CPC advertising deals lies in which party is responsible for accounting and payment. In a CPC referral deal, website A typically is responsible for keeping track and paying for the number of referrals sent to it by R. In the case that R is a search engine, and A is an advertiser in a CPC ads deal, the search engine R is typically responsible for keeping track of and charging for referrals are made to A.

Finally, in general, referral deals can be action-based instead of just click-based. That is, A may pay R not just for clicks, but instead for page views or sales made on A's web site. Also, referral deals often involve less complexity than advertising deals in which online auctions typically take place to determine the web site to which a user may be referred.

3 Types of Spam

An online ad network can be abused in many ways, and we will describe a few of them in this section. Each of the ad revenue models described in the previous section (impression-based, click-based, and action-based) can be subject to corresponding types of spam. Respectively, we discuss impression spam, click spam, and conversion spam in this section.

Fraud based on such abuse can result in reduced ROI for advertisers, although there are many other reasons that advertiser ROI can suffer. For instance, advertiser ROI can suffer due to low quality ads, high latency at the advertiser's web site, improperly targeted keywords, non-competitive sale prices, and many other factors. Nevertheless, fraud is one of many such variables that can account for non-optimal advertiser ROI. In this section, we provide an abridged taxonomy of different types of "spam" that may impact an online ad network.

Spam is an artifact that does not provide any value, utility, or benefit as expected by one or more parties associated with the artifact. For instance, an email is often considered *spam* when the recipient of it does not derive any appreciable value, utility, or benefit from the email. In the following subsections, we will also discuss click, impression, and conversion spam.

3.1 Impression Spam

We already mentioned impression spam in Section 2.1, and we briefly expand upon the impact that it has on CPM-based advertising and CTR calculations in this subsection.

Impression spam results from HTTP requests for web pages that a user never sees or provide advertisers little or no value for their money. Impression spam affects CPM-based advertising campaigns because advertisers should not be charged for such HTTP requests.

Furthermore, impression spam affects click-through-rate (CTR) calculations, as the denominator in the calculation is the number of page views. Since the ranking of ads in an auction may depend upon CPC and CTR, manipulating CTR can be of interest to a malicious advertiser as an indirect method to manipulate the outcome of the ad auction. In addition, click fraudsters may end up increasing the CTRs of ads that they click on, and higher-than-normal CTRs can serve as evidence of a click fraud attack. As such, click fraudsters may conduct impression spam to decrease CTR on ads that they click on in an attempt to avoid detection.

Once HTTP requests that are deemed to be impression spam are filtered out of a web server log, the number of total page views minus those that are deemed to be impression spam should be used in CTR calculations as well as for CPM counts.

3.2 Click Spam

Click spam is a type of spam which occurs when sources other than legitimate users are making HTTP requests for ad impression URLs. Such HTTP requests are often called "invalid clicks." *Invalid clicks* are any clicks that an ad network chooses not to charge for. When clicks are marked invalid, the user agent that issued the click is still directed to an advertiser's web site, but the advertiser is simply not charged for the click.

A *fraudulent click* is one that was issued with malicious intent, and *click fraud* is simply the practice of issuing fraudulent clicks. Since intent is only in the mind of the person issuing the click or in the mind of the author of software that issues clicks, it is impossible to know with 100 percent certainty that any click is fraudulent. While the intent of a click is impossible to determine, there are various signs or *signals* that indicate the intent of the click with varying certainty. Clicks that are believed to be fraudulent are marked invalid, but it is not necessarily the case that all fraudulent clicks will be detected and marked invalid.

When suspected fraudulent clicks are marked invalid, the user agent is still redirected to the advertiser's web site. Proceeding with the redirection offers two benefits. Firstly, a fraudster does not receive any feedback as to whether or not she has been detected as such. Secondly, if a suspected fraudulent click is, in reality, a legitimate click (a false positive), then the user's experience is not negatively impacted, and may provide the opportunity for increased ROI for the advertiser. That is, a user's click may be marked invalid, and yet the user may go on to purchase a product on the advertiser's site! However, too many false positives may decrease a publisher's revenue, and hence an ad network must do its best to minimize false positives to balance the trade-off involved in providing high advertiser ROI while concurrently attracting and maintaining quality relationships with publishers.

Note that, in many cases, invalid clicks are not necessarily fraudulent, and may have nothing to do with malicious intent. Clicks may be marked invalid, for instance, due to an ad network's desire to improve advertiser ROI. Many such clicks that are marked invalid are due to double clicks (e.g., a user that clicks twice to open a link), crawlers that ignore `robots.txt` directives, and other technical reasons.

Two sources of invalid clicks that do occur due to malicious intent are advertiser competitor clicking, and publisher click inflation. In advertiser *competitor clicking*, a malicious advertiser may click on a competitor's ads. There are many reasons that a malicious advertiser may engage in competitor clicking. One possible, simplistic intent is to drain the competitor's advertising budget. Once the competitor's advertising budget has been drained, the malicious advertiser's ads may be exclusively shown to a user (assuming that there are no additional advertisers bidding for the same keywords). More generally, an advertiser may want to decrease competition for a particular keyword. By fraudulently clicking on other's ads for that keyword, competitor's derive less ROI, and expend their budgets. In the short term, the auction becomes less competitive as competitors

become ineligible to participate due to reaching their daily budget limits. In the long-term, due to the decreased ROI for such keywords, competitors may reduce or stop spending, and CPC rates for the fraudster will be lower.

When the AdSense was introduced by Google, publisher click inflation became an additional type of click fraud that needed to be mitigated. In publisher *click inflation*, a publisher clicks on ads on her own site in the hopes of receiving a revenue share of the CPC that the advertiser pays for ad clicks. The incentive for fraud in publisher click inflation is a more financially direct one than in the case of advertiser competitor clicking.

While we have briefly discussed some incentives for click fraud in this section, we discuss click fraud countermeasures in Section 5.

3.3 Conversion Spam

A *conversion* occurs when an HTTP request is issued for an advertiser-defined URL. For example, the request of the URL could signify a page view, submission of a form, initiation of a file download, or the completion of an online sale.

Conversion spam occurs when HTTP requests are issued with the intent of artificially producing conversions. Just as there is incentive for click spam in CPM-based advertising to make it appear that ads are relevant via inflated CTRs, there exists an incentive for conversion spam in CPC-based advertising to make it appear that clicks are relevant. For instance, after clicking on an ad and being redirected to an advertiser's site, a click fraudster may download a file (such as a trial version of a software package) from an advertiser's site in an attempt to simulate the behavior of a legitimate user.

CPA-based advertising is also susceptible to conversion spam, as fraudulent publishers might like to derive a revenue share from artificially produced actions. For example, if such a publisher gets paid a revenue share from a CPA-based advertising campaign based on downloads of a trial software package, that publisher may initiate such downloads herself after fraudulently clicking on ads in an attempt to derive revenue from the self-initiated downloads.

4 Forms of Attack

Click fraud can take on many different forms. Nevertheless, there are some key technology and resource limitations on what vectors an attacker may try. For example, someone with little time and low determination will only attempt simple attacks from their home computer. On the other hand, a dedicated fraudster with more time may write malware, infect machines worldwide, and command the infected machines to click on ads. In this section, we describe some representative forms of attack; we do not attempt to comprehensively enumerate all known forms of attack, but we merely seek to give the reader some familiarity with some common forms of attack seen "in the wild."

The space of attacks against networks fall into two main categories: human, where people manually click on ads, and robotic, in which traffic is generated

programmatically by software robots. We describe each of these categories of attack in more detail below.

4.1 Human Clickers

Humans are harder to manage than automated software robots, and are fundamentally limited by the speed at which they can think and act. Nevertheless, hiring humans to click on ads is a feasible vector by which to attack an ad network in geographies in which human labor is particularly cheap. Even in developed economies, humans can be coerced or leveraged to click on ads just as they can be coerced into solving CAPTCHAs in exchange for the privilege of viewing pornographic images [GRS05]. From an economic point of view, using humans for click fraud purposes is viable as long as the cost of directly paying or coercing a human to click on an ad is lower than the returns of clicking on the ad.

For publishers that are paid a share of ad click revenue on their web site, these costs can be quite low. For example, if it takes 10 seconds for a fraudster to view a page and click on an ad with a return of $0.10 USD, a fraudster can realize returns of $0.60 per minute, or up to USD $36 an hour for clicking on ads. Thirty six dollars is a reasonable daily salary in many parts of the world at the time of writing of this text. By reducing the incentives and slowing fraudsters down to 1 cent every 20 seconds, a person can still make $1.80 an hour, or approximately $2500/year, which is roughly the per-capita GDP in some developing countries. Even accounting for connectivity costs (which can be $10/month) and computer costs (cheap computers can run $300-$400), a person can theoretically make a decent living being paid to defraud advertisers, if countermeasures are not in place. (We discuss countermeasures in Section 5.) Of course, constantly loading pages and clicking on ads can be rather tedious.

Given that cheap sources of Internet-connected labor exist, fraudsters have generated several attacks that take advantage of humans. Such attacks all have similar goals, but have different costs and degrees of implementation difficulty. These human-intensive attacks can involve only one human or multiple humans whose efforts can be coordinated. Multiple numbers of machines and/or IP addresses can be taken advantage of. In addition, various means of acquiring the machines and IP addresses can be used. However, before discussing such targeted, manual attacks, we first discuss how publishers might encourage or coerce legitimate users to unintentionally click on ads. Such ad clicks may or may not be fraudulent, but neither the advertisers' nor the users' interests are being respected. While much of the later sections in this chapter deals with the technology used to conduct and contain click fraud, the next subsection deals with the psychology of the user.

4.1.1 Coercion

Good website design is focused on helping a user accomplish a task, and is often guided by interaction designers and usability engineers. For example, a web

storefront may be designed and implemented to help a user to find what she is looking for and purchase it in an efficient fashion. A social networking site may be designed to help users find and interact with others that have similar interests. On the other hand, in an ad-centric website design, a web publisher may attempt to design a site to "encourage" visitors to click on high CPC ads.

To ensure that high CPC ads will appear on her web site, a publisher can replace or supplement regular web page content with collections of keywords for which advertisers typically place high bids. This practice is sometimes called "keyword stuffing." The keywords can be hidden from the user by including them in HTML tags that are not visibly displayed, or by rendering the keywords in the same color as the background of the web page. In both cases the "stuffed" keywords will be seen by the ad network when its automated agents crawl the page to determine relevant ads to show, but not by a human user viewing the web page.

To now encourage users to click on the high CPC ads, the publisher does not include very many (if any) regular links to other web pages, and attempts to make any regular links look like ads as well. A user who then wants to click a link to go to another web page may inadvertently click on one of the high CPC ads because it is not clear how to easily navigate to a more interesting web page. Similarly, a publisher might modify ad text to something more relevant to coerce users, while retaining the original advertiser URL (e.g., replacing "Click here for Viagra" with "Free Ringtones"). Or, a publisher may go so far as to make ads invisible to users and give them directions about actions to take (e.g. "hit tab twice, then enter") in order to force them to unknowingly click on ads.

Other devious publishers may host games, often written in Macromedia Flash, on web pages such that the game is surrounded by ads. Part of the game play may involve clicking and dragging the mouse, and by encircling the game with ads, users tend to unintentionally click on ads, providing a near-term benefit to the publisher.

At one extreme, a web publisher explicitly asks users to click on ads, for example by including "call-to-action" text on a web site stating "please click on the ads on this page." At another extreme are websites that are intentionally designed and engineered to coerce users to click on ads, even without an explicit call-to-action. Regardless of the technique used, all of these schemes aim to deceive users and provide poor user experience. Moreover, users often end up at a site they had no intention of visiting and in such cases, deception leads to low advertiser ROI.

4.1.2 Manual Clicking

A malicious user can generate a relatively large volume of traffic even over a relatively short period of time. Even a person who is not very Internet savvy can generate thousands of page views and hundreds of ad clicks over the course of a day. For a website that earns revenue based on CPM or CPC ads, such page views and ad clicks can translate to hundreds of dollars a day in revenue. Moreover, an advertiser that wants to drain a competing advertiser's budget

can potentially rob the competitor's budget of hundreds of dollars a day if appropriate countermeasures are not in place. On the other hand, such behavior can be easily detected if the same user is returning to the same site again and again. Such a manual clicking attack can be simple to execute, but can also be relatively simple to detect. When such manual clicking attacks appear at high volumes, fraudulent intent can be obvious as it is rare for a single user to generate large amounts of targeted page view and ad click traffic during normal browsing of the web.

To make such a manual clicking attack harder to trace and detect, some fraudsters use HTTP proxies to obscure the source of their clicks. HTTP proxies can anonymize traffic by acting as an intermediary between a user's machine and web site. Proxies can hide the source IP address, and strip identifying information, such as cookies, from HTTP requests. HTTP proxy services are often available for free or at nominal fees. Some attackers also set up a means to systematically re-route traffic through a series of proxy machines to further obscure the original source of HTTP requests. Anonymous routing systems and distributed networks of machines such as Tor [DMS04] sometimes provide a means for a fraudster to anonymize their traffic, although such systems can also be used legitimately for anonymous web browsing. In addition to stripping identifying information from HTTP requests, a given user's traffic may become hard to uniquely identify because that user's traffic can be "mixed in" with many other users' requests arriving from an HTTP proxy. HTTP requests for ad impression URLs (clicks) that emanate from such networks can be viewed as suspicious by ad networks. One might argue that a legitimate user might want to take advantage of such anonymization services, but at the same time if the value to advertisers of clicks emanating from such services is dubious or traffic from such a service is anomalous for other reasons, an ad network may decide to mark such clicks as invalid.

An attacker can also explicitly coordinate groups of users to click on ads to drain advertiser budgets or attempt click inflation. In one scenario, a *click farm* firm can be hired to click on ads. Such a firm may be profitable in a third-world or developing nation, and may hire employees to click on ads in its offices or may coordinate contract workers that click on ads from cyber-cafes in different cities.

Naturally, distributing such fraudulent ad clicks across many malicious users and machines increases the burden on the detection systems of ad networks by forcing a network to aggregate activity over a period of time and across many IPs to find significant sources of fraudulent activity. Interestingly, activity distributed as such can be difficult for a single advertiser to detect because the activity lies "below the radar" for a given advertiser but can be detected using aggregated traffic across multiple advertisers. Such distributed attacks can take a significant amount of effort to coordinate and are indicative of fairly determined fraudsters.

An interesting case study of a manual, distributed, click fraud attack occurred on the Google search network in 2006 on clicks relating to the the keyword *kiyashinku* (cashing *credit* or *credit-cards* in Japanese). Google's Ad Traffic

Quality Team noticed an increase in query and click traffic on the keyword that emanated from large numbers of seemingly legitimate users. It turns out that an inflammatory blog post was found that incited readers to issue a search query for the keyword on google.com, and then click on the resulting ads in an effort to drain the advertising budgets of Japanese credit card vendors.

The resulting traffic of the cashing attack looked very anomalous as there was a sudden surge in interest in these high-value ads. Moreover, users were clicking on the same ad dozens of times, or clicking on large numbers of ads returned as a result of the query. In cases like these, though, user interest quickly wanes and the attack is short-lived.

Humans looking to supplement their income can also be recruited directly to click on ads by "pay-to-read" (PTR) and "pay-to-click" (PTC) websites. Such sites accept membership registrations from users (usually other than publishers). The "Pay-To" web site sends the users instructions on what sites to "read" or click on in return for an extremely small share of revenues derived from such activity. Such sites give explicit directions to users on web pages or in emails about how to act in an attempt to ensure activity appears legitimate from these sites. For example, instructions might specify to "stay on the website for at least 30 seconds", or "click on links that are interesting to you."

However, users are often fundamentally uninterested in the content of these sites, and are often more interested in the prospect of being paid for their browsing and clicking. One can find some differences between programs paying users to surf and programs paying users to click on ads, since surfing does not guarantee ad clicks, but many of these sites seek to indirectly generate revenue from CPM- and CPC-based advertising. PTR/PTC programs are also often part of a pyramid scheme in which some "users" pay into the program to receive better pay-outs. As with most pyramid schemes, most users rarely if ever receive any significant payouts themselves.

An intriguing effect of many of these sites is that networks of "pay-to" sites are often all linked to one another via ads. Effectively, users who visit pay-to sites are likely to sign up on other pay-to sites. One pay-to-click website may show advertisements on other pay-to-click sites in an attempt to attract other like-minded users.

Despite the advantages of using humans to conduct click fraud and otherwise generate invalid clicks, people can be hard to convince, hard to provide incentives to, and can get bored or tired quickly. Moreover, a person paid to view pages or click on ads can act in ways distinctly different from a user truly interested in, say, purchasing a product. For instance, a real user tends to read, consider, think, and surf a website because she wants to learn more about the product. A paid user has few such interests. In addition, PTR and PTC users might be less likely to conduct purchases or "convert" and the lack of conversions can be visible to advertisers in the form of ROI. There are significant gray-areas in what an ad network needs to consider fraudulent – bad behavior on one network may be normal on another. At the end of the day, fraud needs to be defined by what advertisers expect from services provided by an ad network and normal user behavior.

4.2 Robotic Clicking

Fraudsters often give up any pretense of legitimacy in their click inflation efforts, and resort to automated traffic generated by software robots, or "bots". Bots carry one major advantage over human users – they do what they are told, over and over again, and they do it for free, without needing direct motivation. Running a bot can require less coordination and avoids the pesky requirement of finding and interacting with people.

Many software robots are written for legitimate purposes, such as scraping websites or crawling web links. Sites may place robot exclusion files (i.e., `robots.txt`) in specific directory trees to direct automated crawlers to avoid scanning parts or all of a website. Advertising networks often use `robots.txt` files to prevent robots from inadvertently crawling ads.

A malfunctioning or improperly configured bot may not abide by the `robots.txt` directives and unintentionally click on ads. These bots are not intended to be malicious, and make little effort to "cloak" themselves. They often announce themselves as robots through their user-agent string or other fields in the headers of their HTTP requests, making it easy to detect and account for inadvertent bots.

In contrast, some bots are constructed for the sole purpose of clicking on ads. These are referred to as "clickbots" [DStGCQT]. A *clickbot* is a software robot that clicks on ads by issuing HTTP requests for advertiser web pages with the intent to commit click fraud. Clickbots can be custom-built or purchased, and can be manually installed by fraudsters or disseminated in a manner similar to malware and worms.

If appropriate countermeasures are not in place, large clickbot networks could potentially conduct significant amounts of fraud, even if each bot executes just a few ad clicks per day. Moreover, the operational costs to run a botnet could decrease over time, entailing even lower risk for attackers [JBB+07].

Unfortunately for clickbots, their predictability is a significant weakness. Developing a bot that behaves like a human and cloaks its behavior is challenging. An ad network can use hundreds or thousands of signals that may indicate if HTTP requests might have been generated by a machine instead of a human. Despite this, robotic traffic sources are used by attackers to send invalid traffic to advertisers. Building a robot to click on ads does require some skill as a programmer and also requires significant insight into how one might write a software bot to look and act like a human.

Some costs involved in using bots include authoring or purchasing bot software, and obtaining a network of machines to run the bot software on. Often, these machines have been illicitly compromised and are rented for hire. Current research indicates botnet rental fees run in the cents-per-bot-week range [lan, JBB+07].

4.2.1 Custom-Built Bots

There are many tools available that may be used to develop custom-built bots. Open-source utilities such as *wget* [Nik] and *curl* can, for instance, be used to issue HTTP requests. Shell scripts can be written to invoke such programs in an automated fashion as one form of attack. Libraries that help programmers execute HTTP requests exist for many common programming languages such as Python, Java, C/C++, and Perl. Some bots can also be written as Browser Helper Objects (BHOs) to leverage existing functionality in web browsers to help bot authors construct HTTP requests. These open-source utilities, application libraries, and BHO-based methods often also support the use of HTTP proxies, which can help mask the original source of an HTTP request. Such tools can be used to build clients that conduct advertiser competitor clicking or publisher click inflation attacks.

4.2.2 "For-Sale" Clickbots

After a bot is built, it can be sold on the open market. While there exist some legitimate, testing-oriented applications of bots, they can just as easily be used to conduct attempted click fraud. "For-sale" clickbots such as the Lote Clicking Agent, I-Faker, FakeZilla, and Clickmaster can be purchased online. They typically use anonymous proxies to generate traffic with IP diversity. Fortunately, IP diversity usually is not enough to hide click fraud attacks conducted by such software, and traffic generated by them is identifiable.

4.2.3 "Malware-Type" Clickbots

Malware-type clickbots infect machines in order to achieve IP diversity, and their traffic may or may not be as easily identifiable as that generated by for-sale clickbots. Malware-type clickbots can receive instructions from a botmaster server as to what ads to click, and how often and when to click them. Clickbot.A, which is described in more depth in "The Anatomy Of Clickbot.A" by Daswani, et. al. [DStGCQT] provides a detailed case study of such a clickbot investigated by Google, and was published with the hope of helping the security community build better defenses.

4.2.4 Forced Browser Clicks

"Forced browser clicks" is a technique that fraudsters can attempt to use to turn a legitimate user's browser into a clickbot. The technique takes advantage of flaws in some implementations of AdSense-like programs that could be used by a malicious publisher to conduct click inflation. In AdSense-like programs, web publishers are provided with a "snippet" of HTML code that they place on their web pages in locations that they would like ads to appear. The HTML code often contains elements that render ads, and some implementations of the HTML code can be vulnerable to a forced browser click attack. For instance, if the HTML code simply contained an anchor element that makes up the ad

(i.e., `<A HREF>`), a publisher could insert additional HTML code on its web page that would instruct the browser to click on the anchor element without explicitly being requested to do so by the user.

To implement dynamic ad slots securely, the ad slot code can take advantage of the "same-origin" security feature [DKK07] implemented in most browsers to prevent script from the publisher's parts of the web page from clicking on the ads. For instance, an HTML IFRAME tag accepts a SRC attribute whose value can be set to an URL owned by the ad network. Using the IFRAME construct, the browser will not allow script on other parts of the web page to access the contents of the IFRAME if that content was not downloaded from the *same-origin* as the IFRAME's content. The origin is specified by the domain name, port, and protocol in the URL. Google's AdSense ad slots are implemented as such to protect against fraudulent publishers, though not all ad networks are equally protected.

However, it is worthwhile noting that even when an ad network uses a construct such as the IFRAME from allowing other, untrusted parts of the web page to interact with its ads, the ad network is relying on the browser and its implementation of the same-origin security model to defend against publisher click inflation. If a widely deployed browser had an implementation vulnerability that allowed a publisher to circumvent the same-origin model, then the ad network may be vulnerable to publisher click inflation.

The forced browser clicks technique was first described publicly in a research paper on "Badvertisements" [GJR06], and Chapter **??** of this book is based on that research paper. The interested reader is referred to that chapter for more details on forced browser clicks.

5 Countermeasures

This section describes representative countermeasures that an ad network might deploy to mitigate the click fraud risk to an online advertising business. State-of-the-art click fraud mitigation takes advantage of "defense-in-depth." [DKK07] Many countermeasures and defenses are used, so that the probability of a successful attack is reduced by each layer of countermeasures a fraudster must circumvent. We divide our discussion of click fraud countermeasures into methods that are targeted at prevention, detection, and containment.

Before proceeding with our discussion of click fraud countermeasures geared at prevention, detection, and containment, we note that the goal of such countermeasures is to "raise the bar" of conducting a successful attack to the point where the expense and effort of an attack outweighs the potential reward (monetary or otherwise). As such, there is no claim that the countermeasures we describe are perfect or eliminate click fraud in any absolute sense. Rather, the goal of click fraud countermeasures is to greatly lower the risk of a successful attack to advertisers, publishers, and online advertising networks, and to support a profitable system for all parties involved (except, of course, the fraudsters). It is also not sufficient to introduce only enough countermeasures to allow ad-

vertisers to profit from an online advertising ROI that is better than can be achieved offline– minimizing click fraud further allows an ad network to benefit from significantly higher CPCs and stronger profitability for itself, publishers, and advertisers.

Finally, we remark that the countermeasures that we describe in the following subsections may not necessarily all be directly targeted at identifying fraudulent clicks, but focus instead on identifying anomalous expected or aggregate user behavior, browser behavior, or other inconsistencies that may be indicative of click fraud. In some cases, clicks associated with such anomalies are marked invalid, whether or not they can be directly linked to fraud.

5.1 Prevention

An old adage says "An ounce of prevention is worth a pound of cure." While it might indeed be impossible to prevent someone from manually clicking on a single ad, it is possible to take steps that can prevent larger-scale, more systematic click fraud from occurring.

In the case of click fraud due to publisher click inflation, such fraud can be prevented by exercising care with regard to which publishers might be allowed to participate in an AdSense-like syndicated ads program. In addition, publishers that are terminated for "low-quality" or fraudulent ad clicks sometimes try to re-apply to be included in such programs. As such, it is important for an ad network to uniquely identify publishers, and prevent publishers who have already been terminated from re-joining. Many such publishers attempt to sign up for participation with different, invalid, or fraudulent names, addresses, and/or telephone numbers in an attempt to appear distinct from their previous, blacklisted identity.

Ad networks may look at many different characteristics of HTTP query, click, and conversion traffic that might potentially indicate fraudulent click activity. If fraudsters were aware of what characteristics ad networks look for, they may be able to artificially construct HTTP request patterns that do not exhibit any of the anomalous characteristics that the ad networks look for. While an ad network can be quite open about what types of countermeasures and processes it has in place (see [Tuz06] for an example), it is extremely important for ad networks to maintain the confidentiality of these characteristics, or *signals*, that may be indicative of fraudulent click behavior. Companies that run ad networks must employ the best enterprise security practices available to protect these signals, just as a company would protect trade secrets against both insider and external attacks. Not only are such signals useful in identifying potential click fraud attacks, but they can also be useful for marking low-value clicks as invalid, thereby giving advertisers better ROI, and giving the ad network competitive advantage.

A more concrete preventative technique involves setting up a trust boundary between a publisher's web page content and the ad slots on the publisher's page, as discussed in Section 4.2.4 using an HTML IFRAME. An additional example of a concrete, preventative technique would be for an ad network to

set a maximum CPC which can prevent a fraudster from making significant amounts of money with just a few clicks. Fraudsters who then want to attempt to make significant amounts of money are then forced to produce enough clicks to result in statistically significant signals. However, since such preventative techniques are not perfect, it is important to complement them with detection and containment countermeasures.

5.2 Detection

The purpose of click fraud detection is to mark clicks invalid after they have occurred, since much of the time they cannot be effectively prevented altogether. Clicks may be marked as invalid both "online" and "offline." When invalid clicks are detected *online*, advertisers are not billed for them, whereas when they are detected *offline*, advertisers are credited for the clicks. Online detection is preferred, as an advertiser's daily budget is not charged for invalid clicks, and the advertiser does not lose the opportunity to have its ads continue to participate in ad auctions that take place that day.

Online detection is not always possible. When invalid clicks are detected offline, advertisers are credited for the clicks at a later time. However, the advertiser may still incur the opportunity cost of not having his or her ads compete in the ad auction for that day if the advertiser is budget-limited with respect to the amount of click fraud. At the same time, some types of click fraud can be more easily detected offline, such as an attack consisting of just a few clicks per day that requires many days of traffic in aggregate to identify the attack. Such attacks are harder to identify in an online fashion when they first start occuring.

Click fraud detection methods usually seek to identify anomalies in streams of click data. Different types of anomalies can be caught at different times in the processing of clicks. Some anomalies can be apparent in a single click, and such a click can be marked invalid by an online filter. Other types of anomalies may only become apparent in looking at an aggregate set of clicks over some time period. The decision as to whether or not a particular click fraud detection filter should be implemented as an online or offline one is often guided by exactly what anomalies need to be identified and the time period that needs to be analyzed to identify a particular anomaly. After filters have been applied and invalid clicks have been identified, additional actions may be required. For instance, an AdSense publisher with a high percentage of invalid clicks, or even just one that received potentially anomalous traffic that may or may not be invalid may be inspected by an ad traffic quality investigator. In certain cases, a relationship with a publisher may be automatically terminated if unusual proportions of invalid clicks are attracted by the publisher's web site. As such, ad traffic quality investigators need to also be mindful of potential sabotage that may occur in which a malicious publisher attempts to generate a large number of invalid clicks on a competing publisher's web site to make it seem as if the competitor is attempting a click inflation attack.

Anomaly-based detection can sometimes suffer from data limitations. If

too little data is available for clicks associated with a particular advertiser or publisher, it may be hard or impossible to make a determination as to whether or not some clicks should be marked invalid. If there are too few clicks, the damage due to a click fraud attack may be generally low, so long as the CPCs involved in the attack are not too high. Such an attack can still be identified by analyzing longer time periods or clustering clicks with similar properties. In the case in which the clicks are for high CPC ads, the traffic can be scrutinized even more.

In other cases, if there is too much click data associated with an advertiser or publisher, it may be hard to identify potentially fraudulent clicks as they could get "lost in the sea" of valid clicks. In such cases, the amount of monetary damage due to click fraud may be relatively low compared to the aggregate number of clicks. Nevertheless, identifying chunks of click data associated with advertisers and publishers that are "just the right size" is an important aspect of the problem. In particular, click data can be cut into finer chunks based on various properties to reduce the possibility of fraudulent clicks getting lost in the sea.

In addition to "passive" detection by identifying anomalies in streams of click data, an ad network can conduct "active" detection by modifying the interaction with the web browser and/or the user to generate additional data that can be used to ascertain the validity of a click or a set of clicks. For example, an ad network may decide to add extra Javascript variables in ad slot code that runs computations on clients. Upon receiving ad clicks, the results of such computations are sent back to the ad network, and the ad network can check if the expected results may fit an expected distribution. While it may be apparent to a fraudster that such computations are taking place, the expected result distribution may not be quite as apparent. In such an example, the ad network modifies the interaction with the user agent by requiring it to send back some computation results that may serve as a signal as to the validity of a set of clicks by legitimate users using a web browser (as opposed to, say, a custom-programmed bot that does not support Javascript). While we use Javascript-based computations as an example of an active detection technique, we note that there are many technical details that we have not adequately covered here for using such a technique to effectively serve as an active click fraud detection signal.

5.3 Containment

Top-tier ad networks employ hundreds or thousands of passive and active signals to help detect click fraud. Nevertheless, it is typically hard to ascertain the validity of a click in any absolute sense, even with a plethora of such signals. As such, prudent ad networks may take the stance that some attempted click fraud may not be directly detected, and it is important to contain or manage the potential monetary impact of such clicks to advertisers. In this subsection, we briefly describe two representative click fraud containment measures: smart pricing and manual reviews.

Due to the difficulty in ascertaining whether or not any given click is indeed fraudulent or not, some ad networks charge an advertiser only a fraction of the CPC for certain ad clicks. Google's version of this feature is called *smart pricing*. Based on various factors that can be indicative of advertisers' ROI, such as CTR and average CPC, the overall quality of click traffic can be assessed and captured by a smart pricing multiplier.

A smart pricing multiplier can help an ad network contain undetected click fraud attacks. Even though it may be hard to determine if any click in particular is fraudulent, aggregate characteristics of click traffic may reveal that "something is fishy" and this "fishiness" is captured by a smart pricing multiplier, sometimes also called a Click Cost Multiplier (CCM). It is important to note that the benefits of smart pricing go well beyond containing the effect of any undetected click fraud. For example, a smart pricing multiplier can also take into account the effect of aggressive placement of ads on web pages in which the ads are the most prominent feature on the web page. Even though CTRs may be higher for such ads, conversions might be relatively lower, and a smart pricing multiplier can protect advertisers from being charged the full CPC for clicks that are less likely to convert. In addition, a smart pricing multiplier can also be used to contain poor ad targeting or other forms of poor content quality. Finally, in contrast to techniques we have discussed thus far that attempt to make a binary decision about the quality of a click, smart pricing allows click quality to be scored on a continuum.

Manual reviews of click traffic can also be an important part of an ad network's containment strategy. If some fraudulent clicks evade existing automated detection mechanisms, they can be caught by highly trained engineers and operations personnel that, over time, gain a keen sense of "smell" and intuition. In some cases, manual review requests (either generated by internal processes or by advertisers) could potentially result in identification of a new type of click fraud attack. In such cases, engineering and operations personnel may often be able to generalize the new pattern of attack and develop automated countermeasures to detect similar attacks in the future.

Feedback from advertisers can also be valuable since such feedback can sometimes uncover undetected click fraud due to "holes" in detection systems. As such, feedback from particular advertisers can be used to provide better ROI for all advertisers and benefit the ad network. However, in practice, the majority of inquiries by individual advertisers end up not uncovering undetected click fraud, but instead uncover issues relating to non-optimal ad campaign management or advertiser site design. In a few cases, advertisers correctly identify fraudulent clicks that have already been detected as invalid, and the advertiser was not charged for it. In even fewer cases, advertisers actually do help uncover undetected click fraud.

To provide some quantitative data, Google reports that less than 10 percent of all clicks consistently are detected as "invalid." [Off07] These clicks include both fraudulent clicks and redundant or inadvertent clicks, and advertisers are not charged for these invalid clicks. All these clicks are *proactively* detected as invalid by Google. In contrast, all advertiser inquiries to Google lead to less than

0.02 percent additional clicks being reactively classified as invalid and credited back to advertisers.

6 Click Fraud Auditing

Click fraud auditing is the process of examining the set of clickthroughs delivered to the advertiser's site to determine which of the clicks, if any, may have been fraudulent and/or issued with malicious intent. While an ad network may examine such clickthroughs as part of its click fraud detection process, a company that serves as a trusted third-party (separate and distinct from the ad network and the advertiser) can do such an examination as well.

An analogy between click fraud auditing and auditing of corporate finances can be drawn. Once a company prepares its financial statements, including its cash flow statements, profit and loss statements, and balance sheet, an auditing firm can verify these statements by analyzing the data and receipts used to construct the statements. The auditor can, in many cases, request to see additional data, and the company and the auditor can work together to resolve any discrepancies that are found to ensure that the company is correctly reporting its financial health to the market.

In the case of a click fraud audit, an advertiser is interested in determining if an ad network is correctly reporting the number of valid clicks for which it is charging. The advertiser can hire a third-party organization to serve as an auditor. The ad network and the auditor could work together to resolve discrepancies between the number of valid and invalid clicks indicated by the ad network's click logs and an advertiser's web logs. Click fraud audit reports, when constructed correctly, have the potential to:

- provide value to advertisers by giving them confidence that they are "getting what they are paying for,"
- allow click fraud auditors to serve as a trusted-third party who can help arbitrate discrepancies, and
- help ad networks identify and fix limitations in their click fraud detection and containment systems.

When done incorrectly, click fraud audit reports can provide misleading information to advertisers and cause them to negatively impact their business by altering advertising campaigns that would otherwise result in cost-effective conversions.

The process of click fraud auditing involves several challenges, including maintaining the confidentiality of signals, overcoming data limitations, and maintaining user privacy.

6.1 Confidentiality of Signals

As part of a click fraud audit, an ad network cannot disclose information about exactly which clicks it is has charged for and which it has marked invalid because

that would leak information about its signals [2]. After all, a malicious advertiser could experiment with issuing a single or a small number of clicks on its own ads each billing period to determine how many of those clicks are deemed valid. Each set of clicks that the malicious advertiser issues could be targeted at testing a hypothesis that the advertiser might have about a specific, surmised signal that the ad network might be employing. If an advertiser repeats such experiments many times, it may effectively "reverse engineer" an ad network's signals, and then use the information to conduct competitor click fraud, sell the information, or issue a click inflation attack on a site for which it is also the publisher. As such, an ad network cannot disclose exactly which clicks it marks valid or invalid as such information could be used to create a "playbook" for attackers about how to circumvent signals.

6.2 Data Limitations

While click fraud auditing companies are typically given information about the set of clicks that arrive at the advertiser's web site, they may be at a disadvantage as compared to ad networks with regards to being able to assess which clicks should and should not be considered valid. Since ad networks cannot disclose information about exactly which clicks they do and do not charge for, a click fraud auditing company is left to take a guess about which clicks the ad network may or may not have considered valid. A click fraud auditing company may attempt to make its own assessment about which clicks should and should not be considered valid by looking for anomalies. For instance, if a click is issued on the same ad by the same IP in an advertiser's web log multiple times, the click may look anomalous to a click fraud auditing company if the only information they have is the advertiser's web log. At the same time, an ad network has information on clicks from extremely large numbers of advertisers, and a click that may look anomalous in the small stream of data that a click fraud auditor has from the advertiser may look completely normal in the ocean of data that an ad network has from, say, hundreds of thousands of advertisers. For instance, the multiple clicks on the same ad from the same IP in the previous example may be completely valid if that IP is a HTTP proxy for a large ISP such as America Online or a mobile carrier. A large ad network benefits from seeing click data from across the Internet, and may be able to do more effective and accurate anomaly detection even compared to a click fraud auditor that has several thousand advertisers as clients.

In addition to being able to use a much larger amount of click stream data in which to look for anomalies, ad networks have web logs from search requests that, when coupled and linked together with web logs containing ad clicks, can effectively and accurately help identify signs of click fraud. All ad impression URLs are generated in response to a query, and, in some cases, cookies are served to users that receive these queries. However, neither the queries used to

[2] For transparency, however, Google does report to advertisers the number of invalid clicks that it discards on a daily basis.

generate ad impression URLs nor the cookies will appear in the advertiser's web logs and hence will not be available to a click fraud auditor. Such query logs are approximately two orders of magnitude greater in size than click logs, and contain a wealth of information that can be mined for the purposes of identifying fraudulent, anomalous clicks that can be marked invalid by an ad network.

Furthermore, in analyzing the click stream data that a click fraud auditor might have from an advertiser's web log, there are effects for which the auditor may not be able to compensate on its own. For instance, after clicking on an ad impression URL, the user is redirected to the advertiser's site, and the HTTP request corresponding to the advertiser's "destination URL" will appear in the advertiser's web logs. If the user continues browsing the advertiser's web site by clicking links to other parts of the advertiser's site, and then hits the browser's back button one or more times, another HTTP request may be made for the advertiser's destination URL to reload the page. Such HTTP requests may look like repetitive ad clicks in the advertiser's web logs, and may look anomalous to a click fraud auditor. An ad network only charges for clicks and not subsequent page reloads. The advertiser will only be charged for the click preceding the first HTTP request for the advertiser's destination URL. In fact, the page reloads do not go through the ad network at all and involve only the user's computer and the advertiser's site. However, an auditor may not have any way of knowing which of the requests in the advertiser's web log correspond to actual ad clicks and which are due to page reload requests. The subsequent HTTP requests can often be misinterpreted as a fraudulent click by a click fraud auditor, when in reality it was simply due to the user navigating back to the destination URL after exploring other parts of the advertiser's site. Such clicks in click fraud audit reports are called "fictitious clicks."

To help remedy the problem of fictitious clicks, Google provides advertisers a feature called "auto-tagging." The auto-tagging feature appends a unique id to the destination URL for each distinct click that has the potential of being charged (i.e., both valid and invalid clicks) to the advertiser's account. The unique id allows click fraud auditing and analytics firms to distinguish unique ad clicks from page reloads. Without this feature, it would be impossible to distinguish page reloads from new, distinct clicks on ads. Due to how commonly users take advantage of such navigation features of web browsers, auto-tagging is essential for proper click fraud auditing. At the time of writing this article, Google is the only ad network providing this feature. More details about fictitious clicks and auto-tagging are provided in [Fcl06].

The various limitations discussed above increase the chances of inaccurate analysis and make it more challenging to do click fraud detection based on advertiser-side data. Hence, advertisers should use care in adjusting their campaigns based on such analysis. Decisions made based on inaccurate analysis can adversely affect sales and can cause advertisers to manage campaigns in a sub-optimal manner. For example, consider an advertiser of leather goods who manages ad campaigns for shoes and handbags in which the advertiser uses two ad creatives, one for shoes and one for handbags, each with separate destination URLs. The advertiser's web logs may indicate several page views for the shoes

destination URL which could be due to reloads caused by users that may have done quite a bit of navigation through the advertiser's site after clicking on the shoes ad. The web logs may not indicate as many page views for the handbags destination URL. An inaccurate analysis of these web logs might tell the advertiser that there is a high level of click-fraud on the shoes ads. However, the page views corresponding to the advertiser's destination URL may actually be indicative of more user interest in shoes due to users' conducting more research about shoes on the advertiser's web site. As a result, it may be worthwhile for the advertiser to invest more in its shoes campaign instead of its handbags campaign in the future. However, if an advertiser invests less in the shoes campaign based on inaccurate analysis, it is unclear as to what the opportunity cost might be; the advertiser might end up losing potential business for its shoes by making a decision based on inaccurate analysis. Furthermore, the missed opportunity on shoe sales could be far larger than any money lost on potentially undetected click fraud. The advertiser is far better served by studying its dollars per conversion rate for both the shoes and handbags campaigns.

It is important for advertisers to closely monitor ROI, and adjust ad campaigns appropriately. Undetected click fraud can manifest itself as lower ROI than expected (based, say, on past performance). At the same time, many other factors such as non-optimal keyword or bidding choices, or a competitor having better sale prices can also negatively impact ROI. Hence, while undetected click fraud will lower ROI, lower ROI does not necessarily imply undetected click fraud is occuring.

6.3 Privacy

One might suggest that if ad networks have more data than click fraud auditors, then perhaps ad networks should share the data with auditors to allow them to help verify the validity of clicks. Unfortunately, to protect the privacy of its users, an ad network cannot share the large amount of data that would be required to allow for such verification.

In addition to considering the number of valid and invalid clicks, it may also be useful for click fraud auditing companies to consider auditing metrics such as the number of dollars an advertiser pays per conversion. Since CPCs are constantly changing due to a dynamic ad auction, and features such as smart pricing are weighting CPCs, the expected number of clicks that an auditor believes should be marked valid or invalid may not be as useful a metric as the expected dollars that an advertiser should pay per conversion.

7 Economics of Click Fraud

As previously discussed, two common forms of click fraud are competitor clicking and click inflation. Competitor clicking is punitive and intended to deplete rivals' marketing budgets. In comparison, click inflation is for-profit and allows fraudulent publishers to collect a share of the click revenue generated by an ad

network. In both cases, ad networks have been criticized as being indifferent to fraud. This is due to a misconception that ad networks suffer no economic consequences from click fraud. In reality, ad networks have strong economic incentives to minimize click fraud.

Ad networks need the trust of advertisers, and have the incentive to provide advertisers with better ROI if they hope to have advertisers increase spending with them in the long-term. Market competition is another key incentive for ad networks to combat click fraud. There are many ad networks actively competing for both advertisers and publishers. Both advertisers and publishers will choose the ad network that offers the best return on investment. Ad networks that offer lower click fraud rates will have a competitive advantage in the market.

Consider the case of competitor click fraud, which saps money directly from advertisers' budgets. Besides violating an advertiser's trust in the ad network, competitor click fraud directly reduces advertisers' ROI. Advertisers victimized by competitor click fraud will either reduce their bid prices or drop out of bidding completely. As a result, there will be fewer participating advertisers. Those advertisers that do participate will bid lower prices for potentially fraudulent clicks which will result in lower competitive pressure in ad auctions, and, in turn, lower profits for the ad network. Minimizing competitor click fraud is in the ad networks' interest, since it will maximize the bid prices that an ad network receives and offer a higher ROI to advertisers who might otherwise use a rival ad network.

The same applies to click inflation. Fraudulent publishers generating bogus clicks on ads displayed on their own properties will reduce the ROI of advertisers. In turn, advertisers will lower their bid prices, and as a result, legitimate publishers will receive a smaller share of ad revenue for their valid clicks. These publishers will choose the ad network that offers the best value for their legitimate clicks. Often, ad networks are also one of the largest publishers on the network themselves. For instance, an ad network may run a keyword search engine, and display ads next to search results. Click inflation in other parts of the network will reduce the bid price that it would receive for clicks on their own properties.

In a competitive environment, ad networks face pressure to reduce both competitor clicks and click inflation. By reducing click fraud, ad networks will improve the returns of advertisers and publishers, as well as their own returns. More efficiently delivering relevant ads to legitimate users will benefit all parties in the system – except the fraudsters.

8 Conclusion

This chapter has provided a summary of the current state of online advertising fraud, including representative forms of attack and representative countermeasures. We discussed some of the challenges involved in click fraud auditing, which, when done correctly, can even provide advertisers that receive high ROI further confidence in online advertising. In addition, we have commented on the

economics of click fraud and the incentives that search engines have to catch as many fraudulent clicks as possible to maximize advertiser ROI.

There is much art and science still being developed surrounding various aspects of mitigating online advertising fraud because the online advertising market is in a state of expansion while working to support the needs of advertisers and online commerce providers. Successful fraud management will provide competitive advantage to ad networks, and help enable them to provide the highest ROI possible to advertisers.

9 Acronyms

BHO	Browser Helper Object
CCM	Click Cost Multiplier
CPA	Cost-Per-Action
CPC	Cost-Per-Click
CPM	Cost-Per-Mille (Impressions)
CTR	Click-through Rate
eCPM	Expected Cost-Per-Mille (Impressions)
HTTP	HyperText Transfer Protocol
PTR	Pay-To-Read
PTC	Pay-To-Click
PPC	Pay-Per-Click
ROI	Return-On-Investment

10 Acknowledgements

We thank the editors, Markus Jakobsson and Zulfikar Ramzan, for the opportunity to contribute this chapter. The authors are grateful to Tom Dillon, Thomas Duebendorfer, Carl Evankovich, Daniel Fogaras, Tri Le, and Razvan Surdulescu for providing feedback that helped improve the quality of this chapter.

References

[DKK07] Neil Daswani, Christoph Kern, and Anita Kesavan. *Foundations of Security: What Every Programmer Needs to Know*. Apress, 2007.

[DMS04] Roger Dingledine, Nick Mathewson, and Paul Syverson. Tor: The second-generation onion router, August 2004.

[DStGCQT] Neil Daswani, Michael Stoppelman, the Google Click Quality, and Security Teams. The anatomy of clickbot.a. http://www.usenix.org/ events/ hotbots07/ tech/ full_papers/ daswani/ daswani.pdf.

[Fcl06] How fictitious clicks occur in third-party click fraud audit reports. http://www.google.com/adwords/ReportonThirdPartyClickFraudAuditing.pdf, August 2006.

[GJR06] Mona Gandhi, Markus Jakobsson, and Jacob Ratkiewicz. Badvertisements: Stealthy Click-Fraud with Unwitting Accessories. *Journal of Digital Forensic Practice*, 1(2):131–142, 2006.

[GRS05] Craig Gentry, Zulfikar Ramzan, and Stuart Stubblebine. Secure distributed human computation, 2005.

[JBB+07] Collin Jackson, Adam Barth, Andrew Bortz, Weidong Shao, and Dan Boneh. Protecting browsers from dns rebinding attacks. *ACM CCS*, 2007.

[lan] Secure grid computing: An empirical view. http://www.laas.fr/IFIPWG/Workshops&Meetings/48/WS1/10-Landwehr.pdf.

[Nik] Hrvoje Niksi. GNU Wget. *available from the master GNU archive site prep. ai. mit. edu and its mirrors*.

[Off07] Official AdWords Blog. Invalid clicks - google's overall numbers. http://adwords.blogspot.com/2007/02/invalid-clicks-googles-overall-numbers.html, February 2007.

[RAM98] Michael Reiter, Vinod Anupam, and Alain Mayer. Detecting Hit Shaving in Click-Through Payment Schemes. *Proceedings of the 3rd USENIX Workshop on Electronic Commerce*, pages 155–166, 1998.

[Tuz06] Alexander Tuzhilin. The lane's gifts v. google report. http://googleblog.blogspot.com/pdf/Tuzhilin_Report.pdf, July 2006.