How Fictitious Clicks Occur in Third-Party Click Fraud Audit Reports

Click Quality Team
Google, Inc.
August 8, 2006

Summary
Google has detected pervasive reproducible problems in the way third-party click fraud auditing firms gather and report their data. These problems cause their reports to contain fictitious clicks – clicks which were never made on Google AdWords ads. Because of these fictitious clicks, third-party click fraud auditing firms significantly overestimate the number of clicks occurring on an advertiser's account – and even more significantly overestimate the amount of "click fraud" detected. This report presents:

- Background on third-party click fraud estimates and methodology problems
- Findings from an internal Google review of third-party click fraud auditing reports
- Recommendations for addressing this issue
- Demonstrations of how fictitious clicks occur in third-party systems
- Detailed case studies for three major third-party auditing services

Background
We have seen numerous reports of click fraud estimates which we believe significantly overestimate the impact on advertisers. We examined and addressed the methodology of one of the widely quoted reports a few months ago here: http://adwords.blogspot.com/2006/03/about-invalid-clicks.html.

We have continued to see third-party click fraud auditing firms stating that their measurements show much higher levels of click fraud than we believe could possibly be realistic (e.g. 14%), which is troubling. These estimates have also had a high variance, ranging up to 35%. Although only a very small number of AdWords advertisers use third-party click fraud auditing firms, the click fraud estimates produced by these firms have been highly publicized and caused concern for many advertisers. We also receive click fraud auditing reports generated by these firms from advertisers requesting refunds.

The most fundamental flaw that we have seen in these reports is the existence of *fictitious clicks*: events which are reported as fraudulent but do not appear within Google's logs as AdWords clicks. This report identifies the root causes behind these fictitious clicks and illustrates the extent to which this flaw impacts click fraud estimates from these firms.

Over the last year, these estimates have received widespread media coverage. A different kind of report (from Outsell, Inc.) has also been widely cited for estimating the scope of the problem. But in fact that report did not measure click fraud – it was an opinion survey

of advertisers asking them to guess at the extent of the problem. Thus the report's conclusions about the percentage of fraud and financial loss for the industry are essentially a poll of the *perception* of the size of the problem (with the backdrop of the previous coverage of high estimates) rather than actual *size* of the problem. This is analogous to estimating crime rates in a country by asking some residents how much crime they think there is, and averaging those guesses to state that number is the actual rate.

We have been aware of the presence of fictitious clicks in third-party reports for some time. We have given feedback to advertisers (and indirectly, to some of these third-party auditing firms) and pointed out the various flaws we've observed in their reports, but have met with little in the way of a positive response or interest in correcting their methodologies. They maintained that their click fraud detection methodologies differed from ours, and that fact alone accounted for any differences.

We are also concerned that advertisers may be hurting their businesses by changing their campaign settings based on such erroneous information – for example, by reducing spend on an otherwise successful campaign, keyword, or channel that, according to these reports, appears to have high levels of click fraud.

During the past several months, we have conducted careful analysis to identify the root causes for fictitious clicks across several click fraud auditing firms.

The results were troubling and unexpected. We discovered some basic engineering and accounting issues across the industry – problems which were in fact completely separate from the issue of accurate click fraud detection – which have in each case led to dramatic overestimation of click fraud rates by these firms. As an example, a single AdWords click may appear as five events in some reports, leading to (a) the identification of these events as "click fraud", and (b) the reporting of five fraudulent clicks.

We have presented our initial findings and some case studies here. It is our hope that this information will compel firms in this industry to invest more heavily in R&D to provide a sound engineering-based approach. We do understand the desire of some advertisers to use third-party auditing services, and we will work with these firms and with industry groups to establish standards that will hopefully lead to more accurate reports for these advertisers.

Findings

The major root causes for fictitious clicks fall into the following two categories:

- **Fictitious clicks due to detection of page reloads as ad clicks**. This is the counting of page reloads on an advertiser's site as multiple clicks on the advertiser's AdWords ad – which did not actually occur. Page reloads can occur for various reasons, including:
    - user browses more deeply into the advertiser's site, then hits back button, causing a potential reload of the original landing page

    - user presses browser reload button on the landing page
    - user opens a new window in Internet Explorer, causing a reload of the landing page
- **Fictitious clicks due to conflation across advertisers and ad networks**. This is the counting of one advertiser's traffic in another advertiser's report, even if the advertisers span different ad networks.

These two problems are serious, and have resulted in significant inflation of click fraud estimates from each of the click fraud auditing firms we examined. Appendix A provides simple sequences of events which illustrate these problems occurring in practice.

Additional observations:

- Events identified as fraudulent in these reports, which actually match real clicks in our logs, often converted at nearly the same rate (and in some cases better) compared to other clicks. For example, in one case where 800 paid clicks were marked as "fraudulent", the rate of conversion for these clicks was 5.1%, which compared favorably with the 5.8% overall conversion rate the advertiser achieved on approximately 24000 paid clicks.
- The use of sample-based analysis or the failure to provide the total number of events analyzed (whether they were fraudulent or not) denies advertisers the ability to do even the most basic cross-checking, for example by comparing to the total number of clicks reported by Google. Such a basic test would have alerted them to the presence of fictitious clicks. The fact that third-parties have also not taken measures to do this type of checking raises some serious questions.
- Aside from the above issues, we have also seen basic engineering errors in some of these reports (covered in Appendix B).

Appendix B presents detailed case studies for three firms: ClickFacts, Click Forensics, and AdWatcher. Click fraud estimates from ClickFacts and Click Forensics have received widespread media coverage. And among third-party auditing reports submitted to us by advertisers, reports from AdWatcher are the most common.

All three cases studies exhibit the problem of severe click inflation in their reports primarily due to the presence of fictitious clicks, which generally render their published estimates on click fraud invalid.

Next steps

What Google will do

- Work with third-party click fraud auditing firms to address their engineering and accounting issues
- Continue to provide feedback to advertisers when flawed reports are submitted in order to help them avoid making harmful advertising decisions based on faulty data

- Work with these firms and such industry groups as the IAB Click Measurement Working Group to establish standards in this area, especially with respect to the format of reports submitted to Google (for example, see AdWords auto-tagging instructions in next section)
- Continue our heavy investment in invalid click detection technology, and continue to keep the industry informed about issues related to click fraud

What third-party auditing firms need to do

- Address click inflation issues as soon as possible by making use of all available tools and approaches to count clicks accurately
    - Google AdWords auto-tagging feature (http://www.google.com/adwords/learningcenter/text/31854.html) appends a unique identifier to the advertiser's URL for each click, which for example makes it much easier to distinguish between real ad clicks and page reloads. Google charges at most once for each unique tag (if we determine that the click is valid). We further recommend including these unique tags in auditing reports, which will effectively eliminate the potential for fictitious clicks. This feature is automatically turned on for advertisers using Google Analytics (https://adwords.google.com/support/bin/answer.py?answer=31216), and since November 2005 has been available for all AdWords advertisers.
- Invest in R&D and engineering, and take quality issues seriously
    - We do believe that there is a place for third-party click auditing firms in the industry's value chain, but only for those who can deliver real value to all stakeholders. In fact we would value having true vulnerabilities in our invalid click detection systems pointed out to us. However, to date, we have not yet discovered a single legitimate vulnerability as the result of a third-party click fraud auditing report.
- Avoid sample-based analysis, and provide full information to advertisers, such as the total number of events from Google that were analyzed, so that advertisers can do independent checking

What advertisers can do

- Track your ROI carefully
    - The number of conversions (sales or leads) produced by your AdWords campaign is the ultimate proof of its success
    - Tools such as Google Analytics enable you to further optimize your site and advertising campaigns
- If you are concerned about click fraud and want to know more about how Google detects invalid clicks, review the information we have provided here:
    - http://adwords.blogspot.com/2006/03/about-invalid-clicks.html
- If you want to track the number of invalid clicks detected (and filtered out) on your account, use our AdWords invalid clicks reporting tool:
    - http://adwords.blogspot.com/2006/07/estimating-invalid-clicks.html

- For advertisers wishing to use a third-party click auditing firm, we recommend a careful selection process, and adhering to at least the minimum guidelines we outline here

Appendix A: Problem Demonstrations

The following two sections provide simple user activity sequences to illustrate how the "page reload" issue and "conflation of events across advertisers and ad networks" arise. We used real advertiser sites, but have anonymized them in the sequences to avoid spurious visits to their site; we plan to share the non-anonymized sequences with the third-party auditing firms presented in this report. The issues illustrated are in no way caused by any of the advertisers, but are inherent in the way the third party auditors gather their data and logs.

A1. Page reload problem (affects all third-party auditing systems examined)

Here is an example of how incorrect disambiguation can happen by 3rd parties (or advertisers). The following sequence was produced using Internet Explorer.
The sequence below involves an advertiser using ClickFacts, but similar sequences can be produced with advertiser sites that use Click Forensics or AdWatcher.

If a user does a search for "********" on Google, and clicks on the ad for ********.com, the advertiser's server gets the following request from the user's browser with the following referrer information:

1)
Request: http://www.********.com/index.php?comp=googaw&cmptype=********
Referrer: http://www.google.com/search?hl=en&q=********

and the user's browser also sends the following request/referrer to the ClickFacts server as part of loading the advertiser's page (because of code inserted on ********.com's page):

2)
Request:
htttp://track.clickfacts.com//tracker_files/track.gif?J=1&X=http%3A//www.google.com/search%3Fhl%3Den%26q%3D********&url=http://www.********.com/index.php?cmp=googaw&cmptype=********&siteid=29&F=1&C=1&h=2
Referrer: http://www.********.com/index.php?cmp=googaw&cmptype=********

Now assume the user clicks on the "contact" link on ********.com's site. The user's browser sends the following request/referrer information to ********.com:

3)
Request: http://www.********.com/contact.php
Referrer: http://www.********.com/index.php?cmp=googaw&cmptype=********

Now assume the user presses the "Back" button on their browser as part of their browsing activity. In this scenario, because the advertiser's original landing page is dynamically

generated, the user's browser cannot use a cached copy of the page and sends a request to the advertiser's server to get the contents of the original landing page again (note that this request is no longer the result of a click on a Google ad). This effectively results in a "page reload":

4)
Request: http://www.********.com/index.php?comp=googaw&cmptype=********
Referrer: http://www.google.com/search?hl=en&q=********

And the browser also sends the following request to ClickFacts again as part of reloading the advertiser's original landing page:

5)
Request:
http://track.clickfacts.com/tracker_files/track.gif?J=1&X=http%3A//www.google.com/search%3Fhl%3Den%26q%3D********&url=http://www.********.com/index.php?cmp=googaw&cmptype=********&siteid=29&F=1&C=1&h=4
Referrer: http://www.********.com/index.php?cmp=googaw&cmptype=********

Note that the combined Request/Referrer information in (1) and (4) look identical to the advertiser, and (2) and (5) look identical to ClickFacts. This shows that simply looking at the request and referrer information leads to event (4) or (5) to be incorrectly identified as arising from a Google ad click. The potentially counter-intuitive behavior by the browser is that the referrer information in (4) is identical to the information in (1), making it difficult to distinguish the initial request caused by the Google ad click from the follow-on request caused by a page reload. Correctly determining that the latter events are not caused by a Google ad click and are the result of subsequent user browsing behavior requires a more complex analysis. For example, advertisers could possibly analyze their logs to realize a user came from Google, went deeper in their site, and their subsequent request for the homepage is likely the result of user hitting the back button on their browser. In the above case, ClickFacts could not even do this analysis with their current data gathering methodology since they do not have access to advertiser logs and depend on their own logs, and as shown above, their servers do not register the fact that the user went deeper into the advertiser's site before generating the second request (5) to ClickFacts.

The above misinterpretation of server events/logs results in event (4) or (5) being incorrectly attributed to a Google ad click when no such ad click took place on Google. And the sequence makes it appear as if two Google ad clicks happened in close proximity when there was really a single ad click involved.

We want to emphasize that the above ambiguity arises because of the way browsers work in general, and is not particular to how Google ads work. We encourage advertisers and third-parties to use the Google AdWords auto-tagging feature described in the "Next steps" section earlier (a feature that has been available since November 2005) to make it much easier for third parties to avoid the above serious mistake.

A2. Conflation of clicks across advertisers and ad networks in AdWatcher

AdWatcher obtains traffic logs either through a redirect mechanism (where an ad click goes to AdWatcher's server before being redirected to the advertiser's site) or through adding code to the advertiser's web pages. AdWatcher also maintains a cookie that keeps track of the frequency of visits for a given user.

The following sequence shows that this frequency count actually keeps track of the user's behavior across different advertisers and across different ad networks. Furthermore, it counts more than just ad clicks, and gets incremented as the user browses within an advertisers website. The AdWatcher system works by detecting fraudulent activity when this frequency count exceeds a certain threshold. On the event that causes the count to exceed the threshold, the affected advertiser is alerted with the perception that all of the events behind that count were fraudulent events on that advertiser's account. This leads to an immense amount of double-counting across advertisers and across ad networks.

Therefore, in addition to inflating click counts similar to the previous "page reload" sequence, AdWatcher also attributes ad clicks on other ad networks such as Yahoo, and ad clicks on other advertisers in the number of fraudulent clicks that it reports to an advertiser. So it is not uncommon for a single Google ad click to be portrayed as tens of fraudulent events.

We used Internet Explorer to generate the following sequence, with all cookies cleared at the start. The frequency count mentioned above is labeled as "ses_fc1" in AdWatcher's cookie below. The sequence below shows that 1 Google ad click on Advertiser A, followed by browsing on Advertiser A's website, followed by 1 Yahoo ad click on Advertiser A, followed by 1 Google ad click on Advertiser B yields a frequency count ("ses_fc1") of 6. If Advertiser B were to set his threshold for this count to 6 to detect suspicious traffic, the report generated by AdWatcher would make it appear as if the single Google ad click for advertiser B was actually 6 fraudulent events from that IP on advertiser B's ads.

1. User finds advertiser A's site on Google and clicks on the ad (1 Google ad click for Advertiser A)

```
Search on Google:
http://www.google.com/search?sourceid=navclient&ie=UTF-
8&rls=GGLD,GGLD:2005-05,GGLD:en&q=*******+*****+********

User clicks on the ad, leading to the advertiser's site:
http://www.******************.net/

AdWatcher's site receives a request as part of loading the
advertiser's webpage, and sets the AdWatcher cookie as follows:

Set-Cookie: ses_fc1=1; expires=Monday, 06-Aug-07 21:50:12 GMT;
path=/;domain=.adwatcher.net
```

```
Without any additional interaction from the user, the frequency
count gets incremented a second time by AdWatcher before the page
has completely loaded! So a single Google ad click has already
led to a frequency count of 2.

Set-Cookie: ses_fc1=2; expires=Monday, 06-Aug-07 21:50:12
GMT;path=/;domain=.adwatcher.net
```

2. User clicks on the "enter" link to go deeper within advertiser A's site:

```
http://www.*****************.net/introduction.html

AdWatcher's site receives a request as part of loading this page,
and increments the frequency count again:

Set-Cookie: ses_fc1=3; expires=Monday, 06-Aug-07 23:06:26 GMT;
path=/;domain=.adwatcher.net
```

3. User presses the browser “Back” button causing a page reload:

```
This leads to a second loading of the advertiser's top page at:
http://www.*****************.net/

The loading of this page causes another increment of the
AdWatcher frequency count:

Set-Cookie: ses_fc1=4; expires=Monday, 06-Aug-07 23:10:24 GMT;
path=/;domain=.adwatcher.net
```

4. User finds advertiser A's site on Yahoo and clicks on the ad (1 Yahoo ad click for Advertiser A)

```
Search on Yahoo:
http://search.yahoo.com/search?p=*******+*****+********&fr=FP-
tab-web-t500&toggle=1&cop=&ei=UTF-8

Clicking on the advertiser's ad leads to following url:
http://www.*****************.net/?OVRAW=*******%20*****%20*******
*&OVKEY=*******%20*****%20********&OVMTC=standard

As part of loading this page, AdWatcher increments the frequency
count:

Set-Cookie: ses_fc1=5; expires=Monday, 06-Aug-07 23:13:05 GMT;
path=/; domain=.adwatcher.net
```

5. User finds advertiser B's site (not related to advertiser A) on Google and clicks on the ad (1 Google ad click for Advertiser B)

```
Search on Google:
http://www.google.com/search?hl=en&q=********
```

This advertiser uses the redirect feature of AdWatcher, hence this request gets redirected through AdWatcher to the advertiser's url:

http://www.**********.com/******.htm

The AdWatcher frequency count gets incremented as part of getting to the advertiser's site:

Set-Cookie: ses_fc1=6; expires=Mon, 06 Aug 2007 23:16:18 GMT; path=/; domain=.adwatcher.net

Appendix B: Case Studies

Note: the data presented in the case studies are anonymized to maintain the privacy of both advertisers and users.

B1. ClickFacts Case Study
http://clickfacts.com

The following analyzes three ClickFacts reports sent to us by the same advertiser, the first two reports covering the same time period during February 2006, and the third report covering a period during June 2006.

ClickFacts adds extra code on the advertiser's landing page to gather traffic logs for their analysis.

ClickFacts Report #1 (period of time during February 2006)

Google's information about this advertiser for the above time period:

- # of valid (charged) clicks (from Google): approx. 24000
- # of invalid (filtered clicks) (from Google): approx. 800 (3.2% invalid clicks rate)
- # of conversions (from Google): ~1400 (3.2% conversion rate)
- A large portion of the invalid clicks were not based on suspicion of malicious behavior, but instead due to double clicks caused by user browsing behavior (e.g., clicking twice to open a link). As such, there were actually 12 conversions generated from some of these invalid clicks which Google did not charge the advertiser for (1.5% conversion rate on the invalid clicks).

The ClickFacts report was a 517 page PDF document which included details of each event ClickFacts had identified as "click fraud". The report concluded with "Fraudulent Clicks Found 2159" (the number of event records provided was actually 2261 even though the summary mentions 2159 events).

The report did not provide the total number of events that were analyzed. The advertiser informed us that ClickFacts claimed a 35% click fraud rate during this period, which implies they analyzed a total of only about 6200 events (so they used 25% of the actual number of Google ad clicks and extrapolated a click fraud rate from that).

The format of a sample record entry in the report was as follows:

```
• Time of Click: 02/04/06 - 04:02:36 PM EST
• Keywords Used: "fiat 124 ****** ********"
• IP Address: 69.**.205.**
• Refer Page: http://www.google.com/search?q=%22fiat 124 ******
```

```
   ********%22&hl=en&lr=&start=10&sa=N
• Landing URL: http://www.********.com/index.php?R=cf
• Reasons for Refund:
   o Same computer clickthrough
   o Repeated IP block & same keyword
   o Repeated clicks from the same IP block
   o Same computer and identical keywords
```

We immediately noticed some glaring problems with the report:

1. There was a massive duplication of events in the report. ClickFacts provided 2261 raw event records. However, these records corresponded to only 445 distinct events according to their own record information. For example, a single unique event was duplicated 12 times, 21 unique events were each duplicated 10 times, 88 unique events were each duplicated 9 times, and so on.

2. For all the times reported for the events in the report, all 2261 events took place at 2 minutes past the hour (i.e., time looked like **:02:** AM/PM EST").

Based on just the above two glaring issues, it was clear to us that this report has major problems without even looking at any detailed data on Google's side.

We found even more serious flaws when we checked the ClickFacts events with ad clicks in Google's logs. The following example is illustrative of the flaws:

- The ClickFacts report showed 54 record entries from IP 69.**.205.** for the query "fiat 124 ***** ********" on 02/04/06 with timestamps ranging from "04:02:18 PM EST" to "04:02:44 PM EST". So their claim was that 54 fraudulent Google ad clicks occurred on that IP in a period of 26 seconds. (As mentioned before, all timestamps in the ClickFacts seem to erroneously report the minute at "02". In comparing the hour of the events to Google logs, it appears that their timestamps are based on CST and not EST as claimed in their report.) We would also like to point out that there were a total of 6 unique timestamps for these 54 events, with each timestamp duplicated 9 times. Google logs showed only 1 click from this IP (throughout the whole month of February) with the above query on 14:08:33 PST (Google timestamp), and we charged the advertiser $0.57 for this click. (Furthermore, throughout the month, we only saw 1 ad click from any IP for the query above. So we are confident that potentially extra ad click did not register with a different IP on our side.)

  Therefore, ClickFacts' assertion would imply a 5400% click fraud for this IP during this time period since for a single Google ad click, they claim 54 fraudulent clicks!

We do not attempt to explain what led ClickFacts to generate reports with such glaring flaws, and refer affected advertisers to ClickFacts for an explanation. However, we were quite troubled to learn that ClickFacts issued a widely covered press release in March 2006, using reports similar to the one described above, and claiming that 35% of the paid

referrals from Google stemmed from "bogus traffic" (http://clickfacts.com/press/washington_post_03_16_06.php).

ClickFacts Report #2

We shared our analysis of the flaws in the first ClickFacts report with the advertiser during June 2006, and the advertiser communicated these flaws to ClickFacts, which led them to generate a second report for the same time period as the first report. This report was sent to us by the advertiser on June 23, 2006, with the following explanation: "ClickFacts have reviewed their data collection and found a few issues, however they contend that overall their analysis is accurate. They have corrected the issues and reran the data."

The second report provided 1278 record entries for fraudulent events and presented the following overall statistics:

- Total Clicks: 5841
- Fraudulent Clicks: 1278
- Fraud %: 22

The format of a sample record entry in the report was as follows:

```
Time (EST)            IP Address        Query
02/20/06-08:37:32PM   24.**.52.**       ********* for sale

Referer URL:        http://www.google.com/search?hl=en&q=********* for sale
Landing URL:        http://www.********.com/index.php?R=cf
Reasons for Refund: o Repeated visits from the same computer
                    o Repeated visits from visitors in the same IP block
using identical keywords
                    o Repeated visits from the same computer using
identical keywords
```

While the second report did not have some of the basic reporting mistakes of the first report, some of the old errors still remained and we discovered some new errors that were more visible now that other errors were removed.

The most important remaining error that was present was that ClickFacts was still attributing events to Google ad clicks when no such clicks existed in our logs. The following is an example:

- ClickFacts reported 4 events from IP 24.**.52.** on 02/20/06, all with the query "********* for sale", with the following ClickFacts timestamps: 08:37:32PM, 08:39:27PM, 08:39:29PM, 08:39:57PM.. Google logs showed only 1 Google ad click from this IP at 18:37:29 PST for the same query (and this was the only click during the whole month of February), and we charged the advertiser $1.14 for this click. So again, the ClickFacts report would imply that we had a 400% click fraud rate (which is impossible) on this IP.

To determine the extent of this problem in the ClickFacts report, we attempted to match events in their report to ad clicks in Google's logs using a very loose matching criteria to increase the number of matches (partial matches of IP addresses, larger time window for event, ...). Out of the 1278 events reported as "fraudulent" by ClickFacts, we could match at most 850 events to ad clicks in Google's logs. This implies an average inflation of events by at least 50% by ClickFacts.

The reason for this inflation is that ClickFacts is not correctly disambiguating between events in their logs generated directly as a result of a Google ad click versus subsequent events that appear in their logs due to user or browser behavior (e.g., “page reload”).

In addition to the inflation impacting the total number of events reported by ClickFacts, a large portion of the event groups were mistakenly identified as fraudulent due to incorrectly attributing events to Google ad clicks which makes it appear as if lots of events occurred in a short period of time (similar to the example shown above where the four events in ClickFacts log seem suspicious, but they only match a single ad click on Google).

Additional information we noticed about this report:

1. Out of the 850 events that loosely matched events in Google logs, 43 events were considered as invalid by Google, and we had not charged the advertiser for them.
2. ClickFacts sometimes incorrectly identifies perfectly legitimate comparison shopping behavior (where a user visits the advertiser two or three times within a span of 10 to 20 minutes) as fraudulent.
3. In the few cases where events identified by ClickFacts matched Google's invalid clicks, we noticed that ClickFacts actually missed the full extent of the problem. For example, on 02/20/06, Google logs showed 16 ad clicks from IP 65.***.176.*** (all between 15:43:08 PST and 15:44:20 PST, or within 28 seconds). This IP turns out to belong to a competitor of this advertiser, and Google marked all 16 as invalid (advertiser not charged). The ClickFacts report only identified 4 of those 16 events on that date, in a case where the attack was real and quite trivial to detect.
4. The 850 events that loosely matched events in Google logs yielded 41 unique conversions (we count at most one conversion event per click even if a click yields multiple conversions). Since we charged the advertiser for only 807 of these clicks, the effective conversion rate for these clicks (identified as "fraudulent" by ClickFacts) is 5.1% which compares extremely favorably with the 5.8% conversion rate that the advertiser achieved on their 24K paid clicks during this time period!

In summary, the most significant error in this report is caused by ClickFacts’ incorrect interpretation of their advertiser’s logs, leading to events being attributed to Google that do not match any ad clicks in Google's logs. The other interesting observation is that the events that actually matched Google ad clicks actually converted and the conversion rate was extremely favorable relative to all clicks that the advertiser was charged for during this time period.

ClickFacts Report #3 (period of time during June 2006)

Soon after we received Report #2, we received a third ClickFacts report from the advertiser on July 11, 2006, this time covering a time period during June 2006.

Before we describe the ClickFacts report, here is the information for the advertiser during this time period according to Google's logs:

- # of valid (charged) clicks (from Google): approx. 26000
- # of invalid (filtered) clicks (from Google): approx. 900 (3.2% invalid clicks rate)
- # of conversions: approx. 2200 (8.5% conversion rate)
- As with the first timeframe, a large portion of the invalid clicks detected and filtered by Google were not based on suspicion of malicious behavior, but were instead due to double clicks caused by user browsing behavior (e.g., clicking twice to open a link).

The format of the third report from ClickFacts was substantially different from the first two reports. It provided substantially less information compared to previous reports regarding the events that were considered bad, and the total number of events that were detailed as fraudulent added up to only 32 (not 38 as mentioned in the summary below). The summary provided was:

- Total Clicks: 184
- Fraudulent Clicks: 38
- Fraud %: 21

Here is a sample record entry provided in the report:

IP	Clicks	Hostname
68.***.81.***	3	*.*.*.cox.net

ClickFacts had change the format of their report to omit detailed information such as date, time, and query. However, despite dramatically less information than in previous reports, there were still errors and methodology issues which were evident:

1. As with previous reports from them, we found that they were still inflating their click counts by attributing events to Google ad clicks which never occurred. Consider the IP 68.***.81.*** shown above, where ClickFacts claims there were 3 fraudulent events. Google logs show a single ad click from this IP for the whole month of June 2006 that occurred at 16:51:59 PST on June 20thand we charged the advertiser $0.70 for it. So again, for this IP, ClickFacts would imply that we had 300% click fraud.

2. In addition to the very serious "fictitious click" flaw, ClickFacts had reduced their sampling to a statistically irrelevant level, sampling only 184 events for an

advertiser that had ~27000 Google ad clicks during this period and extrapolating to a 21% click fraud estimate.

B2. Click Forensics Case Study
http://www.clickforensics.com

We have had several advertisers that have sent us reports from Click Forensics. Our analysis of these reports shows that they suffer from the same key problem we have identified with ClickFacts reports. As with ClickFacts, Click Forensics gathers logs by inserting code on the advertiser's landing page. And just as with ClickFacts, Click Forensics is inflating the number of ad clicks due to an inability to distinguish between user behavior on the advertiser's site and actual ad clicks.

We will cover one of the reports we have received from them as an illustrative example. The other reports had the same errors. The example report covers an advertiser's traffic for a period of time in May 2006.

Here is the sample format of a group of events identified as fraudulent by Click Forensics (as sent to us by the advertiser):

```
Data: Keywords Threat Level View Date Time View Length Host Clicks Per Visitor
outlook ********* High      5/3/2006 14:52 3           217.**.7.***   3
outlook ********* High      5/3/2006 14:52 3           217.**.7.***   3
outlook ********* High      5/3/2006 14:52 3           217.**.7.***   3
```

The above depicts 3 events for the query "outlook *********" for IP 217.**.7.*** all occurring on 5/3/2006 at 14:52. Google logs show a single click from this IP at 7:52:11 PST on 5/3/2006 (Click Forensics time seems to be +7hrs relative to PST), and we charged the advertiser $0.34 for this click. However, the report makes it appear as if we have a 300% click fraud rate (which is impossible) in this case.

In some cases, the fictitious events attributed to Google are mixed in with legitimate ad clicks resulting from comparison shopping. For example, here is a group of 6 events identified as fraudulent by Click Forensics (as shown in the report):

```
outlook 2002 how to share ******** High 5/2/2006 15:50 6 206.***.140.** 6
outlook 2002 how to share ******** High 5/2/2006 15:52 6 206.***.140.** 6
outlook 2002 how to share ******** High 5/2/2006 15:57 6 206.***.140.** 6
outlook 2002 how to share ******** High 5/2/2006 15:57 6 206.***.140.** 6
outlook 2002 how to share ******** High 5/2/2006 15:58 6 206.***.140.** 6
outlook 2002 how to share ******** High 5/2/2006 15:58 6 206.***.140.** 6
```

Google logs show a total of 2 ad clicks for this IP (206.***.140.**) on 5/2/2006, both by the same user, the first one at 08:50:19 PST and the second one at 08:57:23 PST with the advertiser being charged for both. Therefore, according to Google's logs the user visited the advertisers site twice within 7min, which sometimes happens because a user doesn't complete their research in a single visit to the advertiser's site. However, the fact that Click Forensics does not correctly disambiguate events in their logs that correspond

to a Google ad click from events that are caused by follow-on user or browser behavior makes it look like there were a large number of events (6) in this period of time and makes their analysis flag this IP as fraudulent. The report from Click Forensics makes it appear as if we had a 300% click fraud rate (which is impossible) in this case.

Again, as with the ClickFacts case, a simple checking of the number of events Click Forensics attributes to Google versus what our AdWords site shows to the advertiser would make the problem with the reports clear.

B3. AdWatcher
http://www.adwatcher.com

For the small number of AdWords advertisers who send us third-party click fraud auditing firm reports, AdWatcher is by far the most popular. As discussed in Appendix A2, AdWatcher obtains traffic logs either through a redirect mechanism (where an ad click goes to AdWatcher before being redirected to the advertiser's site) or by adding code to the advertiser's web pages. AdWatcher also maintains a cookie that keeps track of the frequency of visits for a given user.

Here is a sample record entry from an AdWatcher report:

```
==============================
IP ADDRESS: 136.***.8.***
==============================
Number Of Suspicious Clicks: 8
Time Of The First Click: June 28, 2006, 3:17 pm
Time Of The Last Click: June 28, 2006, 3:22 pm
Referring URL: google.com
Keyword Query: N/A
```

This implies that the advertiser received 8 fraudulent clicks from Google from IP 136.***.8.*** in a period of about 5 minutes (the above times are likely in EST). Google data shows that there was 1 ad click from this IP throughout the whole month of June, (and also only 1 click from that user from any IP for the whole month of June). The click occurred at 12:07:45 EST with the query "**** ******* Workshops", and the advertiser was charged $4.89 for that click. Adwatcher's report would imply an 800% rate of click fraud for this IP.

Another AdWatcher report had such a huge number of fictitious clicks attributed to Google that the total number of fraudulent clicks reported by AdWatcher was ~12000 for a period in June, while the total number of ad clicks that Google generated for the advertiser was only approx. 6800 (of which approx. 800 had been discarded by Google as invalid clicks, so advertiser was charged for only 6000). Therefore, just the number of fraudulent events claimed by AdWatcher's report during the month of June is 1.75 times larger that the total number of clicks that actually occurred on Google (sum of valid and invalid clicks) for this advertiser!