Click Fraud Detection with Bot Signatures

Brendan Kitts
Applied AI Systems
Seattle WA USA
bkitts@appliedaisystems.com

Jing Ying Zhang, Albert Roux, Ron Mills
Microsoft Corporation
Seattle WA USA
jingzha@microsoft.com

***Abstract*— Click Fraud Bots pose a significant threat to the online economy. To-date efforts to filter bots have been geared towards identifiable useragent strings, as epitomized by the IAB's Robots and Spiders list. However bots designed to perpetrate malicious activity or fraud, are designed to avoid detection with these kinds of lists, and many use very sophisticated schemes for cloaking their activities. In order to combat this emerging threat, we propose the creation of Bot Signatures for training and evaluation of candidate Click Fraud Detection Systems. Bot signatures comprise keyed records connected to case examples. We demonstrate the technique by developing 8 simulated examples of Bots described in the literature including Click Bot A.**

Keywords—bot; click fraud; fraud; robot; IAB.

I. Introduction

Click fraud bots are arguably the most sophisticated bots online, and employ a variety of strategies to cloak their activities [1-7, 9-11]. The effect of these bots on the online ecosystem can be devastating. Without rapid removal, click fraud bots can transfer vast amounts of money (50 billion dollars per year just from Google for instance) from advertisers to fraudulent entities. This ultimately threatens the fundamental economics of online, as advertisers are forced off auctions, and in general content can no longer be supported by advertising [17].

Combating Click Fraud requires significant investment in resources and large-scale detection systems, as Click fraud bots constantly change and evolve in response to detection [8].

In this paper we discuss one technique that may help to increase the industry's overall effectiveness in identifying and removing Click fraud bots. We propose the creation of bot signatures, which are similar in concept to malware signatures, which could be shared between different white hat organizations in order to more quickly identify and remove fraudulent activity.

II. Previous Work

The only established sharing schemes to-date are the IAB Robots and Spiders list [12] and a variety of IP blacklists that are maintained by a scattering of third party organizations.

A. IP Sharing

IP Blacklist sharing efforts originally grew out of necessity for dealing with Email Spam. However, the same IP ranges implicated in email spam are often also implicated in Click Fraud. Public domain IP Blacklist providers include SORBS, CBL, DSBL, UCE Protect, SpamCannibal. Paid Subscription Services are also offered by Spamhaus, Cymru, SpamCop, Threatmetrix and Quova.

B. IAB Robots and Spiders Policy Board

The IAB Robots and Spiders Policy Board is a group which meets monthly to share new bots. The board provides a voluntary protocol for bot developers to identify their bots using a custom user agent string, to respect robots.txt, and to register their bot with the IAB Robots and Spiders list [12] so that the bot activity can be filtered. For instance, this list includes a variety of benign bots including "googlebot", "slurp", "msnbot" and others. The Robots and Spiders list comprises two lists (a) Robots list as identified by useragent strings, and (b) Known browsers. Traffic is filtered if it matches the first list (a known bot) or if the browser is not a "known type" as identified on the second list. This latter list helps to catch bots that are attempting to comply with the self-identification protocol, but which perhaps haven't yet made their way into the list.

However Click Fraud Bots are purposefully designed to avoid detection. As a result, the self-identified useragent protocol above is ineffective when faced with bots designed for fraud.

C. Click Fraud Detection Systems

A further problem for detection efforts is the lack of availability of well-labeled data that can be used to train detection systems. The Tuzhilin report that was developed as part of the Lanes Gifts vs Google Class Action Settlement notes that Google does not maintain positive and negative cases [19]:

"Google does not have full knowledge of which clicks are actually valid and invalid, and it is impossible to identify performance rates of the filters without this knowledge. Still, the Click Quality team could have conducted some studies trying to obtain this knowledge for certain samples of clicks….. Their arguments were that it is extremely difficult to obtain this knowledge in a systematic and unbiased manner for Google. For this reason, Google does not have this information about actual validity of various clicks and, therefore, cannot use the standard TP, FP, TN, FN and other measured described above to determine performance of their online filters."

D. The Need for Bot Signatures

We propose that bot signatures have three significant advantages: (a) they can be shared between white hat organizations, enabling faster elimination of clickfraud threats. (b) they provide an Ad Network with a way to "regression test" their systems and verify that they can filter out bots, and (c) they allow companies to use labeled data to ensure that their system is accurately detecting fraudulent traffic. Using labels, conventional pattern recognition measures such as Area under the ROC curve, can be used to quantify the performance of different classifiers [8], [15], [16], [21].

III. Bot Signature File Format

We propose that a Bot Signature should be defined using two files: (a) Weblog with Case Labels, (b) Case File.

A. Case File

Assuming that an investigator has been able to identify a bot by examining their weblog records, they are invited to create their own CaseID and draft name for their case. Similar to the International Astronomical Union naming convention for celestial bodies, the Investigator could name the bot after themselves and provide them with an intuitive description, eg. "Santy1 bkitts 20080622". Upon review at the periodic Robots and Spiders meeting, naming conflicts can be resolved. The Case Dimension table maintains these records:

(CaseID, Name, Description, Date, Investigator, Class, Event, Notes)

TABLE I. Case File

Case	*Bot Name*	*Desc*	*Date*	*Inv*
1	Santy1 bkitts 2008 06 22	(Details omitted)		
2	Santy2 bkitts 2008 06 22			
3	Scraper1 bkitts 2008 06 22			
4	Santy3 Cookie bkitts 2008 06 22			
5	LWP Bot bkitts 2008 06 22			
6	Double Clicker bkitts 2008 06 22			
7	Santy4 Cookie bkitts 2008 06 22			
8	LF Click Bot1 bkitts 2008 06 22			

B. WebLog with Case labels

The second file needed is a weblog that has case labels added. Bot and fraudulent activity can be fully described by the record of HTTP headers received by a web server: Date, Time, User Agent, IP, Query phrase, Entry Referrer, PublisherURL, Query parameters and so on. As a result, simply recording that HTTP request should be able to fully specify the fraudulent activity. ProbabilityOfBot = 1 if the case is a known bot case, and 0 if it is known to be not.

(HTTP Headers{IP, Useragent, Referrer, Requested URL, LanguageSettings, etc}, CaseID, ProbabilityOfBot, Notes)

1) Valid Auto-Sampling.

In addition to their robotic cases, investigators are also asked to develop a set of "valid" cases as well, which can be paired with the invalid cases. It is important to have valid as well as invalid cases so that it is possible to accurately measure true positives versus false positives.

TABLE II. WebLog with Case Labels

IP	*User Agent*	*Date*	*Time*	*Query*	*Referrer*	*CaseID*	*Prob Of Bot*
A		2/2/2008	1:00:00	PHP	PHP	1	1.0
B		2/2/2008	1:01:00	PHP	PHP	1	1.0
C		2/2/2008	1:02:00	PHP	PHP	1	1.0
D		2/2/2008	1:03:00	PHP	PHP	1	1.0

IV. Bot Strains

We have provided a sample Case Base for the detection experiments in this paper that comprises the following bots:

A. The LWP Bot ("LWP")

Perl programs can often be identified by having a user agent equal to "lwp". This is because the Perl library is named "LWP". We have simulated one of these rogue processes by setting its User Agent String to an obvious value. Case 5 in our Case Base is the LWP Bot.

B. Santy The Search Worm ("Santy1,2,3,4")

Clicks are not the only kind of attack. A good example of an impression fraud worm which infected large numbers of machines is the Santy worm, first detected December 22, 2004. The worm is written in Perl, and when executed, the worm used the Google search engine to look for hosts that have phpBB software in use. It would then directly attack those systems by attempting to exploit a vulnerability in phpBB software to transfer itself to the victim and execute its code [11].

Although Santy's intent was not to disrupt pay per click auctions, it was effective in doing so. When Santy was first detected, it altered the behavior of search from a few million searches per day on keyword "PHP" to several billion. An advertiser who was legitimately bidding on that term would find that their clickthrough rate suddenly dropped to nearly 0, and they were being de-listed.

We have created several "Santy-like" bots which each generate a lot of queries for specific keywords from "infected" users. Some of the bots use cookies, where-as others do not. Santy1 and 2 are basic Santy infections which repeatedly click from an infected user. Santy4 also simulates an infected user with cookies…. Cases 1, 2, 4 and 7 are all examples of Santy.

C. ClickBotA ("CBotA")

On May 19th 2006, PandaLabs reported that it had uncovered a large computer botnet infected with ClickbotA. Ultimately 103,000 computers were found to be infected. The *modus operandi* of the attack was as follows [4]:

1. Machines are infected by downloading a popular screensaver or other methods of infection such as being delivered to an existing botnet.
2. Infected machines pull keywords from a mysql database at random, and fire them against a "doorway" site – a publisher search engine.
3. Doorway site requests ads from Ad Server.
4. Ad server delivers ads back to Doorway site.
5. Infected machine selects a listing from the ad-results at random to click on
6. Infected machine asks Central BotNet Controller whether it "canClick"? It "CanClick" if the Central BotNet Controller has counted less than X clicks against that ad-link in the day
7. Central BotNet Controller also uses a mysql database to store hits per day against ad links
8. Infected machine ceases operation if it has clicked more than X times in the day.

We have created a "ClickBotA simulation". This click bot generates X=1 clicks per user, but attacks from a wide range of users, simulating infected machines. Case 8 in our Case Base is the Low Frequency Click Bot.

D. Search Engine Scraper ("Scraper")

There is a thriving business in checking ranking on Search Engines, and modifying web pages to try to improve those rankings. Sometimes companies will "scrape" search results to find those ranking positions – run a query against a search engine, page down until they find they listing, and then repeat for thousands of queries. This use contravenes Search Engine Terms of Use. We have created a simulated "Search Engine Scraper" by creating a process that simulates doing rank checking. Case 3 in our Case Base is an example of a Search Engine Scraper.

E. Double clicker ("Double")

Not all cases may be malicious bots. Some cases may comprise policy decisions that are enforced by industry groups. There is currently widespread agreement in the industry that double clicks – a second repeated click on the same ad within a certain period of time - cannot be billed. We have set up a case showing double-clicks.

V. Bot Detection Algorithms

We next show how Bot Signatures can be used to build useful quantitative data about the performance of bot detection systems. We describe a subset of methods for bot filtration that are described in other public domain work including [2],[3],[13],[22],[23].

A. User Click Frequency

User Click Frequency is one of the most basic statistical features that can be used for detection. Frequency caps are required by the IAB Click and Impression Filtration Standards [22],[23], and place a limit on the number of clicks that may come from any single user in a certain period of time. If the definition of a user is partially dependent upon IP address then this method is subjected to error because of proxies. An IP which is generating a lot of traffic may be a proxy for a large ISP such as AOL. Mobile traffic is also notorious for using the IP for the carrier.

Despite the problems, frequency capping can be an effective countermeasure for click fraud. In order for fraudsters to generate revenue, they fundamentally need to generate clicks. If the volume of clicks allowed is limited, then the potential damage from an attacker is also limited. This is also why many fraud schemes use distributed attacks.

Frequency capping shows good performance in limiting damage from most attacks. However it fails completely to detect case 8, the low frequency clicker.

B. Presence of Cookie

Most internet users accept cookies during their normal online activities. If a user is not accepting cookies, this can sometimes limit their online experience, and this can be an indicator of bot activity.

In our analysis of historical weblogs, traffic is nearly twice as likely to be bot if it does not have a cookie – the rate of bots in traffic is 3.9% for cookie and 7.8% for non-cookie. Never-the-less, even if the traffic does not accept cookies, still 92% of that traffic is human. In addition, the traffic continues to generate conversions at nearly the same rate as cookie traffic! As a result, cookie alone would result in the loss of

10.4% of known human traffic, and cannot be used to filter traffic.

C. *IP Blacklist*

IP Blacklists are commonly used to identify SPAM senders and other bad sources of traffic. In our application we used an "house blacklist". This blacklist was able to effectively identify case 8, although none of the other cases.

TABLE III. BINARY FEATURES

Metric	Black list	No Cookie
Santy1	0.0%	0.0%
Santy2	0.0%	0.0%
Scraper1	0.0%	100.0%
Santy3	0.0%	2.5%
LWP	0.0%	0.0%
Double	0.0%	0.0%
Santy4	0.0%	0.0%
CBotA	96.4%	98.2%

D. *Clusters of Users*

Yu, Xie and Ke [24] have pioneered techniques to cluster together users that may be part of botnets. We used a prototype version of their method to identify IPs that appeared to be part of a botnet ring.

E. *User Ad Click Sequence Count*

Click Sequence is the number of repeated clicks recorded against an ad. For instance, if the user is clicking for the third time on ad A then we say that their click sequence is 3.

F. *User Keyword Click Count*

This feature counts the number of times a user has clicked on a particular keyword. It is unusual for a customer to repeatedly search for the same keyword.

G. *Results*

Results are shown in Fig. 1 and Table IV. Frequency capping works well on most of the cases but completely fails against case 8 (ClickBotA). As a result, in order to develop a secure click fraud system, this system must necessarily employ multiple features. The above examples illustrate that any single method, such as frequency capping, can be defeated by one or more bot variants.

We tested using a combination approach by training a machine-induced decision tree to utilize the above features in order to determine whether the traffic is bot or human. We used 75% training set, 25% hold-out set. The resulting tree is able to combine superior attributes to detect all of the cases.

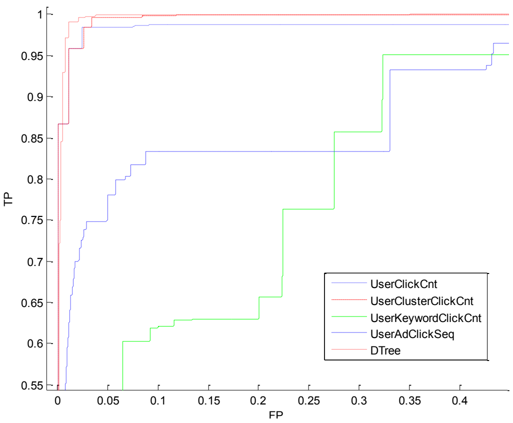

Fig. 1. ROC Curves for five different detection methods against the 8 Example Bots. TP is "true positive" and means that the algorithm flagged the traffic as bot, and it was actually bot. FP means "false positive" and means that the flagged the traffic as bot, and the traffic was actually non-bot. Ideal performance is an algorithm that is able to hug the vertical axis (high true positives and very few false positives).

TABLE IV. AREA UNDER CURVE DETECTION METHOD VS BOT

Bot	***User Click Cnt***	***User Cluster Click Cnt***	***User Kwd Click Cnt***	***User Ad Click Seq***	***D Tree***
All	0.99	1.00	0.86	0.92	1.00
Santy1	1.00	1.00	0.84	0.93	0.99
Santy2	0.99	0.99	0.84	0.94	0.97
Scraper1	0.98	0.98	0.86	0.78	1.00
Santy3	0.97	0.97	0.84	0.94	0.98
LWP	0.95	0.95	1.00	0.95	0.97
Double	0.88	0.86	0.98	0.91	0.92
Santy4	0.93	0.93	0.75	0.86	0.94
CBotA	0.28	0.92	0.38	0.54	0.96

VI. BOT SIGNATURE SHARING

The practice of obtaining and distributing virus signatures is widespread in the anti-virus field. In this field a signature is a characteristic byte-pattern that is part of a certain virus or family of viruses. This byte-pattern may include content of the computer's RAM and boot sectors and the files stored on fixed or removable drives. The creation of these signatures has allowed for easy transmission between anti-virus companies and international researchers, facilitating rapid response to new virus outbreaks.

Sharing of Bot Signatures should be encouraged as a means for the industry to become better at detecting and eliminating bot traffic.

There are two challenges to the realization of this goal. Firstly Ad Networks have an incentive to "Free Ride" by picking up bot signatures by other networks, but not contributing their own. Better Click Fraud detection technology provides a strategic asset which can enable one Ad Network to win market share from another (Mungamuru1 and Weis, 2008) as it increases advertiser value and publisher payout.

A second disincentive to sharing is that there is a security risk from fraudsters infiltrating the sharing companies.

In order to address both problems, we propose that sharing be limited to reciprocal arrangements between trustworthy companies. If a company consistently does not offer bots, then it can lose its membership. An entity such as the IAB would provide an ideal forum for sharing since all companies involved are relatively large and well established.

In addition we believe that a market should be developed where For-Profit companies are able to sell bot signatures in the same way as IP Blacklist subscriptions are currently sold. This kind of market would provide a means for third parties, such as Click Forensics, Authenticlick, and other companies, to become actively involved in participating in detection of bot networks and providing this information to Ad Networks.

REFERENCES

[1] Anupam, V. Mayer, A., Nissim, K., Pinkas, B., Reiter, M. “On the security of pay-per-click and other Web advertising”, *Computer Networks*, Vol. 31, 1999, pp. 1091-1100.

[2] Buehrer, G., Stokes, J. and Chellapilla, K. (2008a) A Large-Scale Study of Automated Web Search Traffic, Proceedings of the 4th international workshop on Adversarial information retrieval on the web (AIRWEB) 2008, April 22, 2008. http://research.microsoft.com/apps/pubs/?id=69505 (2008a)

[3] Buehrer, G., Stokes, J. Chellapilla, K. and Platt, J. (2008b) Classification of Automated Web Traffic, in Weaving Services and People on the World Wide Web, Springer Berlin Heidelberg

[4] Daswani, N., Stoppelman, M. and the Google Click Quality and Security Teams, “The Anatomy of Clickbot.A”, *HotBots 2007: The First Workshop on Hot Topics in Understanding Botnets*, Cambridge, MA, April 10, 2007, http://www.usenix.org/events/hotbots07/tech/full_papers/daswani/daswani.pdf

[5] Edelman, B. The Spyware - Click-Fraud Connection -- and Yahoo's Role Revisited, April 4, 2006, http://www.benedelman.org/news/040406-1.html#e1

[6] F-Secure Virus Descriptions: Santy, http://www.f-secure.com/v-descs/santy_a.shtml

[7] Gandhi, M. Jakobsson, M., Ratkiewicz, J. “Badvertisements: Stealthy Click-Fraud with Unwitting Accessories, Anti-Phishing and Online Fraud”, *Journal of Digital Forensics Practice*, Vol. 1, Special Issue 2, November 2006

[8] Kitts, B., Zhang, J., Wu, G., Brandi, W., Beasley, J., Morrill, K., Ettedgui1, J., Siddhartha, S., Yuan, H., Gao, F., Azo, P., Mahato, R. (2013), Click Fraud Detection: Adversarial Pattern Recognition over 5 Years at Microsoft, unpublished.

[9] Leyden, J., “Click-fraud menace spreads using IM”, *The Register*, October 6, 2006, http://blog.spywareguide.com/2006/10/ie_used_to_launch_instant_mess.html

[10] Leyden, J. “Botnet implicated in click fraud scam”, *The Register*, May 15, 2006, http://www.theregister.co.uk/2006/05/15/google_adword_scam/

[11] Leyden, J., “Worm automates Google AdSense fraud”, *The Register*, May 2006, http://www.theregister.co.uk/2006/10/06/google_adsense_worm/

[12] IAB/ABCe International Spiders & Bots List, Internet Advertising Bureau, http://www.iab.net/iab_products_and_industry_services/1418/spiders

[13] Interactive Audience Measurement and Advertising Campaign Reporting and Audit Guidelines, Version 6.0b, Internet Advertising Bureau, September 2004, http://www.iab.net/media/file/US_meas_guidelines.pdf

[14] Jackson, C., Barth, A., Bortz, A., Shao, W. and Boneh, D. “Protecting Browsers from DNS Rebinding Attacks”, *Proceedings of the 14th ACM conference on Computer and communications security*, October 26, 2007, pp. 421 - 431

[15] Metwally, A., Agrawal, D., Abbadi, A, Zheng, Q. “Hide and Seek: Detecting Hit Inflation Fraud in Streams of Web Advertising Networks”, *WWW*, 2007.

[16] Metwally, A., Agrawal, D. and Abbadi, A. “Using Association Rules for Fraud Detection in Web Advertising Networks”, *Proceedings of the 31st international conference on Very large data bases*, 2005, pp. 169 - 180

[17] Mungamuru1, B. and Weis, S. “Competition and Fraud inOnline Advertising Markets”, in Tsudik (Ed.): *Foundations of Computing 2008, Lecture Notes in Computer Science 5143*, pp. 187–191, 2008.c Springer-Verlag Berlin Heidelberg 2008.

[18] "Pay Per Click fraud botnet discovered”, Help Net Security, May 19, 2006, http://www.net-security.org/secworld.php?id=4002

[19] Tuzhilin, A. *The Lanes Gifts vrs Google Report*, 2006, http://googleblog.blogspot.com/pdf/Tuzhilin_Report.pdf

[20] Troj/Clicker-U Anti-virus listing, Sophos Company website, http://www.sophos.com/security/analyses/trojclickeru.html

[21] Zarokian, P. “Click Fraud – Is it Happening to you?”, *Who’s Clicking Who Report*, October 1, 2003, http://www.clickfraudreport.com/archives/2003/10/index.html

[22] IAB (2005), IAB Impression Measurement Guidelines, http://www.iab.net/media/file/US_meas_guidelines.pdf

[23] IAB (2009), IAB Click Measurement Guidelines, http://www.iab.net/media/file/click-measurement-guidelines2009.pdf

[24] F. Yu, Y. Xie, and Q. Ke, SBotMiner: Large Scale Search Bot Detection, WSDM 2010, February 4 - 6, 2010, New York City.