

Copyright is owned by the Author of the thesis. Permission is given for a copy to be downloaded by an individual for the purpose of research and private study only. The thesis may not be reproduced elsewhere without the permission of the Author.

Analysis, Design and Simulation of  
Fraud and Vulnerability  
Management  
in  
Affiliate Marketing

by

Bede Amarasekara

A thesis

submitted to the Massey University of Auckland

In fulfilment of the  
requirements for the degree of  
Master of Philosophy.

Massey University of Auckland, New Zealand

2016

# Abstract

Affiliate Marketing (AM) is a popular marketing model in e-commerce, which provides businesses a greater reach for a lesser cost. It is considered a safe way to spend the on-line marketing budget, as commissions are paid to affiliates only on monetary outcomes. However, there are inherent risks and frauds associated with the browser-cookie based tracking process. Cookie stuffing, load-time clicking, typo-squatting, conversion hijacking are some of the fraudulent methods used by rogue affiliates to earn commissions for sales transactions that were never actually promoted by them. Some of the previous researches discuss the prevalence of the above frauds, but technical aspects of these frauds, as to how they are implemented and what are the different ways to implement the same fraud are useful questions when developing solutions, which are addressed in this thesis. Contradicting results in quantifying the prevalence of fraud, carried out in previous research work has prompted us to use empirical data to ascertain how widespread these threats are in affiliate marketing. An affiliate marketing dataset of a practitioner spanning over a period of more than four years were analysed. Some of the above fraud scenarios were discovered and the prevalence of fraud scenarios verified. This thesis also presents new vulnerabilities that were discovered using AMNSTE (Affiliate Marketing Network Simulation and Testing Environment). AMNSTE implements same HTTP cookie tracking technology that is implemented in real-world Affiliate Marketing Networks. This simulation and testing environment enables researchers and affiliate marketing practitioners to examine frauds and risk scenarios and to test the efficacy and utility of solutions that are developed to mitigate those vulnerabilities. The thesis finally proposes technical solutions that can be implemented by advertisers and by affiliate networks, as we continue on our ongoing quest to make systems secure from online frauds.

# Dedication

To my parents, in appreciation of their love, commitment and dedication, for never giving up the hope that this token of academic achievement, though long overdue, would one day be true.

# Publications produced

1. Bede Ravindra Amarasekara & Anuradha Mathrani (2015). “Exploring Risk and Fraud Scenarios in Affiliate Marketing Technologies from the Advertiser’s perspective”, Australasian Conference in Information Systems (ACIS 2015). Adelaide, Australia, Dec 2-4, 2015.
2. Bede Amarasekara & Anuradha Mathrani (2016). “Controlling Risks and Fraud in Affiliate Marketing: A simulation and testing environment”, Research paper accepted for IEEE 14<sup>th</sup> International Conference on Privacy, Security and Trust, Auckland, New Zealand, Dec 12-14, 2016.
3. Bede Amarasekara & Anuradha Mathrani (2017 - in-press). “Revenue Fraud in E-Commerce Platforms: Challenges and Solutions for Affiliate Marketing”, Cyber Security and Policy, Auckland, New Zealand: Massey University Press.

# Acknowledgements

I would like to thank my research supervisor Dr. Anuradha Mathrani, for her guidance, support and mentoring throughout my research degree and for inspiring me to reach higher goals.

I thank Associate Professor Chris Scogings for having faith in me and guiding me towards my success.

I also wish to thank my sincere friend Owen Ormsby, for providing me help and support, throughout my study.

Finally, I wish to thank all my family, for being there for me and supporting through all my endeavours.

# Table of Contents

|  |     |
|--|-----|
| Abstract.....  | ii  |
| Dedication.....  | iii |
| Publications produced.....   | iv  |
| Acknowledgements.....  | v   |
| Table of Contents.....   | vi  |
| List of Figures.....   | vii |
| 1 Introduction.....  | 1   |
| 1.1 Research purpose.....  | 2   |
| 1.2 Research goals.....  | 3   |
| 1.3 Thesis structure.....  | 3   |
| 2 Background and related work.....   | 5   |
| 2.1 Technical Overview of an Affiliate Marketing Platform.....                   | 7   |
| 2.1.1 Affiliate Management and Tracking by the Advertiser.....                   | 9   |
| 2.1.2 Affiliate Management and Tracking by an Affiliate Management Platform..... | 9   |
| 2.2 Case Scenario.....   | 10  |
| 2.3 Tracking Process.....  | 11  |
| 2.3.1 Click-tracking.....  | 12  |
| 2.3.2 Conversion-tracking.....   | 12  |
| 2.4 Findings: Risks and Vulnerabilities.....                                     | 13  |
| 2.5 Tracking failures.....   | 17  |
| 2.6 Information security.....  | 18  |
| 2.7 Summary.....   | 19  |
| 2.8 Acknowledgements.....  | 20  |
| 3 A simulation and testing environment.....                                      | 21  |
| 3.1 Design Objectives and specifications.....                                    | 21  |
| 3.2 Description of the Prototype.....  | 24  |
| 3.3 System Architecture.....   | 26  |
| 3.3.1 Affiliate Network platform.....  | 27  |
| 3.3.2 Advertiser’s e-commerce site.....  | 32  |
| 3.3.3 Affiliate’s Website.....   | 33  |
| 3.4 Using the prototype system.....  | 34  |
| 3.4.1 Using AMNSTE to discover frauds and risks.....                             | 35  |
| 3.4.2 Using AMNSTE to find solutions and test efficacy & utility.....            | 37  |

|     |  |    |
|-----|--|----|
| 3.5 | Conclusion and Future Research.....                                      | 37 |
| 3.6 | Summary .....  | 38 |
| 3.7 | Acknowledgements.....  | 38 |
| 4   | Analysing the Data .....   | 39 |
| 4.1 | Use of two Affiliate Networks .....                                      | 39 |
| 4.2 | Tracking failures .....  | 40 |
| 4.3 | Duplicate Cookies.....   | 40 |
| 4.4 | Duplicate IP addresses .....   | 41 |
| 4.5 | Summary .....  | 45 |
| 4.6 | Acknowledgements.....  | 45 |
| 5   | Conclusion.....  | 46 |
| 5.1 | Summary of Thesis .....  | 46 |
| 5.2 | Recommendations for controlling mechanism against frauds and risks ..... | 47 |
| 5.3 | Future Directions.....   | 51 |
| 6   | References.....  | 53 |

## List of Figures

|   |    |
|---|----|
| Figure 1: How to find an Affiliate program to join? (Source: Collins, S. 2011b) ..... | 6  |
| Figure 2: A logical view of the Tracking process .....                                | 11 |
| Figure 3: Layered n-tier architecture .....   | 25 |
| Figure 4: Sequence diagram of the tracking process.....                               | 28 |
| Figure 5. Class diagram of Affiliate Network application.....                         | 29 |



# Chapter 1

## Introduction

E-commerce activities transcend geographical boundaries, opening global markets to developing economies and developed countries alike. Businesses such as those in the Travel and Tourism sector, having customers across the globe, are heavily dependent on e-commerce for revenue generation (Mariussen, Daniele, & Bowie, 2010; Gregori, Daniele, & Altinay, 2013).

There are various ways to attract visitor traffic to an e-commerce site. Websites register on search engines (e.g., Google, Bing, Yahoo), and implement search engine optimization (SEO) with the aim of gaining higher visibility and page rankings. Paid advertising such as Google Ad-words are also very popular among e-commerce practitioners, but without expert knowledge, practitioners can be spending their advertising budget wastefully on an unintended target market.

*Affiliate marketing* (AM) has become one of the main strategies in on-line marketing today (PR Newswire, 2015). It provides businesses a greater reach for a lesser cost, than most other marketing models. AM model consists of four main stakeholders: *Advertiser*, *Affiliate*, *Affiliate Network* and *visitor*. *Advertiser* has a product or service to sell, using an e-commerce site. An advertiser is looking to generate visitor traffic to the e-commerce site by using different advertising models such as AM, organic searches or paid advertising amongst others. *Affiliates* are diverse independent special interest websites and blogs, usually carrying information on a wide range of topics, from travel, nature, motoring, sports and many other subjects, who usually have a large audience of Internet visitors. An *Affiliate* is prepared to display an advertisement of an *advertiser* who sells a product that aligns well with the subject matter of the *affiliate's* website, for a payment. *Affiliate Networks* implement a tracking system, which tracks visitor traffic from the affiliate's website to the advertiser's e-commerce site, on behalf of the advertiser, and determine the payment due for the affiliate for promoting the

advertiser's site (Chachra, 2015; Edelman & Brandi, 2015; Snyder & Kanich, 2015; Dennis & Duffy, 2005; Benediktova & Nevosad, 2008).

### 1.1 Research purpose

In the Affiliate Marketing Ecosystem, many previous research studies have been carried out on fraudulent activities and about cookie-based tracking systems (Chachra, Savage, & Voelker, 2015; Snyder & Kanich, 2015). Edelman and Brandi (2015) have discussed some of the frauds known thusfar and about past litigations that point out the prevalence of some large scale fraud scenarios, but the technical aspects of how those frauds are carried out have not been discussed by the previous authors of AM.

Chachra (2015) developed an advanced browser extension that functions as a web crawler to simulate a real-world user, thus, avoiding detection of its activities, thereby gaining valuable insight into affiliate fraud activity. Snyder and Kanich (2015) re-created browser sessions using HTTP header data from a historical dataset and examined the extent and prevalence of affiliate fraud. A botnet was infiltrated, and traffic analysed by using a Botnet Control & Command (C&C) centre re-writing engine by Kanich, et al. (2008). While these studies have contributed valuable insights into affiliate fraud techniques and have attempted to quantify the fraudulent activities, the authors did not have access to an Affiliate Network (AN) to study the impact of such fraud from within an AN system. These studies were carried out from a publicly visible external standpoint, that is outside of an AM system.

This study has been designed to address the above limitations by examining the frauds and resulting effects thereof from within an AN system, with unrestricted access to each system component. As a result, a scaled-down version of an Affiliate Marketing Platform, that is identical to the real-world cookie tracking system of an AN was developed. The above authors arrived at different conclusions while quantifying the volume of affiliate fraud prevalent in the industry. Chachra (2015) concluded that it is not yet widespread, while Karnich, et al. (2008), Snyder and Kanich (2015) and Edelman and Brandi (2015) concluded otherwise. While different methodologies adopted by above authors resulted in the contrasting

conclusions, a case study undertaken during this research enabled us to have unhindered access to analyse a complete dataset of an AM practitioner to investigate the prevalence of fraud actions, faced by the said practitioner.

## 1.2 Research goals

My research goals are to investigate different methods of carrying out currently known frauds, discover potential risks and vulnerabilities that can lead to further fraud scenarios in future and to find solutions to mitigate those discovered risk factors. To accomplish these goals it required me to:

- a) Develop a simulation environment called *Affiliate Marketing Network Simulated Testing Environment* (AMNSTE), encompassing all the four stakeholders of an Affiliate Network, implementing same technology that is used in a real-world Affiliate Network, which can accurately simulate the tracking processes.
- b) Use AMNSTE to investigate fraud methods in AM, find solutions proactively, and to test the efficacy and utility of the developed solutions.
- c) Investigate tracking data and transaction data of a tourism company which had subscribed to an AM platform, to gain insights on the tracking and process flow of data and to identify possible fraud scenarios.
- d) Attempt to quantify the affiliate frauds prevalent within the industry

## 1.3 Thesis structure

The Thesis first discusses how affiliate marketing platforms operate and gives a brief overview of the technological platform. Next, it explores known frauds and also vulnerabilities which could lead to possible frauds in future. The next chapter presents a simulation and testing platform that was developed in the course of this research that can be used by researchers and AM practitioners alike, to simulate and study the effects of each discovered fraud scenario. Next, a complete click- and conversion-tracking dataset of an AM practitioner is analysed that was generated over a period of 8 months, from August 2013 to March 2014. A second dataset

spanning a much longer period of forty seven months, from April 2010 to February 2014 was also available for further analysis. The second dataset comprises of all conversions that were considered by the third-party Affiliate Network to be successful, legitimate and affiliate generated. The fraud scenarios uncovered during the data analysis are presented next, and solutions for the identified threats are proposed. Some of the solutions which have been developed during the writing of this thesis will be further researched and developed in my next stage of study.

Chapter 2, in part, is the material as it appears in the paper “Exploring Risk and Fraud Scenarios in Affiliate Marketing Technologies from the Advertiser’s perspective” (2015) in proceedings of the Australasian Conference in Information Systems (ACIS). The thesis author was the primary investigator of this paper.

Chapter 3, in part, is the material as it appears in the paper “Controlling Risks and Fraud in Affiliate Marketing: A simulation and testing environment” (2016) that will be presented at the IEEE 14th International Conference on Privacy, Security and Trust. The thesis author is the primary investigator of this paper.

Chapter 4, in part, is the material as it appears in the paper “Revenue Fraud in E-Commerce Platforms: Challenges and Solutions for Affiliate Marketing” (2016) as a chapter of the book “Cyber Security and Policy”, Auckland, New Zealand: Massey University Press 2016. The thesis author was the primary investigator of this paper.

Chapter 5 concludes with the study findings and proposes recommendations on how to mitigate the discovered risk factors, in form of technical solutions, and also discusses the future direction for this important and relevant area of study.

# Chapter 2

## Background and related work

Affiliate Marketing has been around for more than two decades now. CDNow is considered the early pioneer in AM, starting in 1994 (Hoffman & Novak, 2000, Fiore & Collins, 2001), while Amazon.com has the most successful affiliate program with over one million members worldwide (Fox & Wareham, 2012). Venugopal, Das and Nagaraju (2013) traces AM's origin further back in time, to William J. Tobin, the founder of PC Flowers and Gifts launched on the Prodigy Network in 1989, generating more than 6 million dollars by 1993. Dennis and Duffy (2005) recognized a decade ago, the potential of online marketing through AM strategies. Since then the popularity of AM has grown tremendously over the years (Fox & Wareham, 2012; Gregori, Daniele, & Altinay, 2013).

Advertisers are constantly looking to reach out to potential customers, while some of the non-commercial websites already have a large reach of a specific customer base that an advertiser would be looking for. Thus, the AM concept started to make use of this opportunity. Over the years, the AM model has become very popular and rewarding. Some affiliates have started to first look for best-selling products offered by various advertisers through affiliate channels, and then build new websites, on topics relevant to the product, aiming to attract potential customers to their websites (Collins, 2011a). Figure 1 shows the results of a survey conducted by Collins (2011b) among one thousand four hundred affiliates with the question – How do you most often find about an affiliate program and then join?

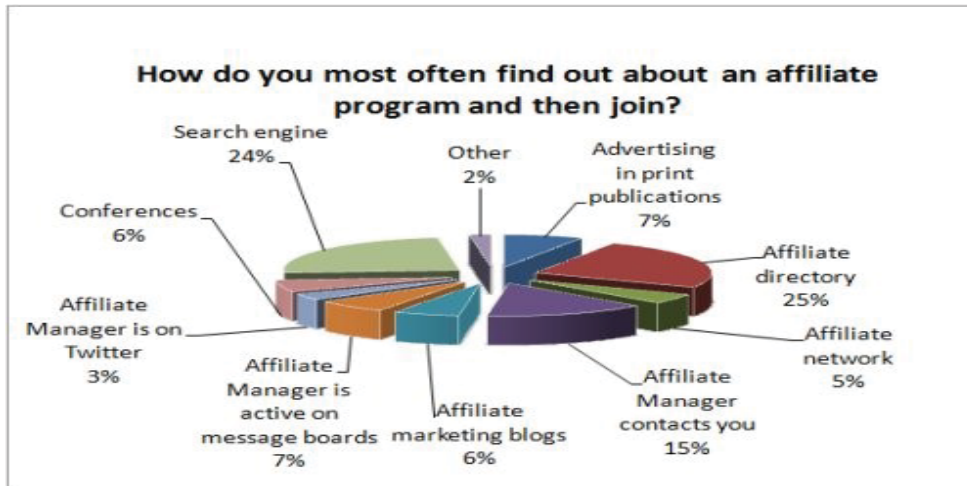


Figure 1: How to find an Affiliate program to join? (Source: Collins, S. 2011b)

AM is considered low-risk and is SME<sup>1</sup> friendly, as it does not require the specialised e-marketing skills that are generally required in payed advertising, which is usually not available to SME's (Fiore & Collins, 2001). It is low-risk, in this context, as commissions are paid only for visits that culminate into sales (Chachra, Savage, & Voelker, 2015). But, there are specific inherent risks associated with AM. Cookie-based frauds allow unscrupulous affiliates to claim commissions for sales that they did not promote (Chachra, 2015; Edelman & Brandi, 2015). Online environments lack visibility and questions are raised over unknown features that may be embedded within the web infrastructure, causing much concern to both businesses and consumers. Usually in AM programs, affiliates and businesses have never seen each other and have a very limited knowledge of each other's businesses and reputations. Trust is an important factor among all the stake holders associated with the AM value chain (McKnight, Choudhury, & Kacmar, 2002; Benediktova & Nevsad, 2008). Advertisers face risks of possible fraudulent activities by affiliates claiming payments for visitor traffic that they did not generate or claiming commissions on sales, for site visits that eventually did not convert to sales. The breach of this trust will cause monetary losses directly to the advertiser and loss of reputation and thereby loss of revenue over time, for the affiliate network. Advertisers expect the affiliate networks to possess the technology to prevent such fraud. Equally, affiliates are wary of advertisers who might wilfully omit some of the sales transactions that should have earned a

<sup>1</sup> Small-to-Medium Enterprise

commission for the affiliate, by deliberately not recording some tracking information, or adjusting sales volumes, etc. At the same time, affiliates trust also depend on whether the affiliate networks have the technical capability to track clicks and conversions, without causing loss of data leading to the affiliate's disadvantage. Finally, the advertisers need to have confidence that the third-party Affiliate Networks they subscribe to, have sufficient security implementations that prevent internal tampering of data, such as, a person with internal access to the affiliate network's databases should not be able to manipulate data to earn wrongful commissions for a rogue affiliate. There has been much academic interest on AM from different perspectives, such as the consumer's perspective (Gregori, Daniele and Altinay, 2013), affiliate's perspective (Benedictova and Nevosad, 2008) and the merchant's or businesses' perspective (Fox and Wareham, 2012; Libai, Bialogorski and Gerstner, 2003; Gregori, Daniele and Altinay, 2013). McKnight, Choudhury and Kacmar (2002) identify trust as the key factor between the consumer and the websites.

## 2.1 Technical Overview of an Affiliate Marketing Platform

An Affiliate Marketing platform connects an e-commerce site of an advertiser with a network of affiliates. Affiliates are third-party, independent websites, such as travel blogs, information pages on diverse subjects and interest groups, etc. who are willing to display advertisement links of different advertisers for a monetary gain. The affiliates will drive consumer traffic to an advertiser's e-commerce site, with the expectation that some of those site visitors might convert visits to sales and become customers of the advertiser. In return, an affiliate earns a commission from the advertiser.

The AM terminology can be confusing and is not always commonly or widely understood, as they can have different meanings under various contexts. A glossary of terminology used by AM and Performance Marketing industry, is compiled by Cake Performance Marketing (Performance Marketing Glossary, 2015).

Usually, managing affiliates and tracking transactions while minimising fraud is a specialised activity which is outsourced to a third-party Affiliate



Marketing/Performance Marketing Management Platform (Edelman & Brandi, 2015). Such specialised platforms have technical capabilities to track transactions across the affiliate network and reporting capabilities that keep affiliates and advertisers up to date on their performance and amounts of revenue and commissions earned. Therefore, advertisers, affiliates, customers and the Affiliate Management Platform (AMP) are four parties that are stakeholders of AM.

Affiliates can measure the volume of traffic their websites have generated towards each advertiser's e-commerce site, by monitoring and logging each click-action by visitors to affiliate's websites. But, if any of those clicks converted to a sale, the value of such sale would not be known to the affiliate, unless the advertiser is willing to implement tracking code, that would notify an affiliate in real-time, of these transactions, as and when the sale occurred. Though the advertiser's tracking code will notify the sales data to the AMP in real-time, it is not the usual case, to notify an affiliates system. Hence, the affiliate relies upon trust with the hope that the advertiser would be honest in declaration of the transactions.

The advertiser may offer the affiliate a Pay-Per-Click (PPC) also known as Cost-per-click (CPC) advertising model, or Cost-per Acquisition (CPA) advertising model, which is currently gaining more popularity. Under CPC, for each visitor who arrives at the advertiser's e-commerce site, by clicking on the advertiser's link on affiliate website, a fixed sum is payed to the affiliate. Under CPA, instead of visits (clicks), only "conversions" (when a visitor has purchased a product offered by advertiser), will earn a percentage of the sale as a commission for the affiliate. The "clicks" under CPC and "conversions" under CPA advertising model, need to be accurately tracked, so that the affiliate who generated the source of the visitor traffic gets paid the commission. *Affiliate* is a specialised third-party technology provider, who offers cookie-based tracking capabilities to advertisers. Among other services, Affiliate Network can recruit suitable affiliates on advertiser's behalf, manage affiliate tracking, affiliate payments and implement numerous fraud-prevention mechanisms to control affiliate fraud, which are prevalent in AM. Alternatively, some advertisers choose to implement the affiliate and tracking management functions in-house, as in the case of large e-commerce practitioners such as E-Bay and Amazon. The fourth stakeholder in AM is, the *Visitor* who is any



visitor to a special-interest website hosted by an *affiliate and* who gets motivated to visit the *advertiser's* e-commerce site.

An advertiser can choose to either manage their affiliate program themselves or outsource this activity by subscribing to an AMP. This is described in the following sub-sections.

#### **2.1.1 Affiliate Management and Tracking by the Advertiser**

Advertisers can maintain logs of the traffic generated by individual affiliates and also logs of sale values that were generated by affiliate driven traffic. The advertiser can make this information available to the affiliates, either through personalised web page dedicated to each affiliate, or by email transaction statements sent periodically (daily, weekly or monthly, etc.). These can be as elaborate as an affiliate portal with dedicated personalised pages consisting of comprehensive reports and charts or be simple periodical transaction statements sent by email. However an elaborate portal would demand a fair amount of web site development and maintenance for the advertiser.

The advantage would be saving the costs of subscribing to a third-party AMP and information security. Ray (2001) states that the perceived advantage of a lower affiliate management cost by running the process in-house, needs a large enough volume of sales. On the other hand, the affiliate has to trust the honesty of the advertiser, as there is no way for the affiliate to verify the data. The lack of trust between affiliates and advertisers causes this to be a disadvantage.

#### **2.1.2 Affiliate Management and Tracking by an Affiliate Management Platform**

The service of specialised third-party platforms are readily available to advertisers, based on a monthly subscription. Outsourcing to these platforms alleviates the need for advertisers to develop and maintain additional functionality to track affiliate activities and the need to design elaborate reporting facilities within their systems. Subscription based AMPs are developed and maintained by external service providers with the latest technologies. Further, trust is one of the core advantages from the affiliate's point of view. Ray (2001) finds the ability of such a platform to introduce new affiliates to an advertiser, with whom they are

already working, an added advantage for the advertiser. But, most of all, the ability to provide unbiased transaction and performance data, which works like a third-party audit, helps to win affiliate's trust, that the data has not been manipulated by the advertiser. This gives a greater credibility to the Affiliate Marketing program of the advertiser. The main disadvantages are the subscription cost and the compromising of information security. The third-party AMP receives all the detailed transaction information of not only the online sales that was driven by affiliates, but every online sale of the company, including direct sales through organic searches, search engines and direct URL visits.

## 2.2 Case Scenario

An investigation of the functionality of an Affiliate Management Network was carried out. An advertiser (a rental car company operating in New Zealand and Australia), a few affiliates connected to this advertiser and a third-party performance marketing/affiliate marketing platform that was used by the advertiser were studied. A prototype has been developed to simulate an Affiliate Network scenario. I have used this prototype to simulate some of the risks and fraud scenarios uncovered during my search, which have been further investigated to build solutions for mitigating these risks. The scenario used in this study is described next.

A car rental company "Best Cars" (pseudonym) has implemented an AM program by signing up with Affiliate Management Platform provider "Connex" (pseudonym). Connex tracks all the visitor traffic to Best Cars that are generated by users clicking on links at affiliate sites. Connex also tracks all visitors who make an actual booking at Best Cars. When a booking is made, Connex calculates the commission amount for the affiliate who generated that click. "Globetrotter" (pseudonym) is a popular travel blog. It has country specific travel pages, which lists many interesting travel tips and latest travel-related offers available in those specific countries. Globetrotter is an ideal web site to become an affiliate for Best Cars, as readers of the country specific travel pages on Globetrotter might be planning a trip to these locations, and might be interested to know of best car rental deals available in these places. Therefore, there are three stakeholders (1)

an advertiser who wants to sell a product, (2) an affiliate who would use his client base (visitors to affiliates web site) to promote the product, and (3) an AMP that uses the technology to track all the traffic from affiliate to advertiser. A typical affiliate network would have many hundreds or thousands of such affiliates promoting the product of the advertiser.

A “Click-Pixel”, which is either an image (banner advertisement) or a text phrase such as “For best deals in car rentals...” within a paragraph of body text, is placed on Globetrotter’s web page, with a hyperlink that points to the Connex Platform. The hyperlink contains affiliate identifier, advertiser identifier, offer identifier and as many data fields an affiliate needs to pass to Connex for tracking, in its request parameters.

On Best Car’s payment confirmation page, a “Conversion Pixel” is placed. The Conversion Pixel is a piece of JavaScript code or in case JavaScript is disabled in the browser, a resource request from Connex web server, with data fields such as session identifier, total price, etc. as parameters. The session identifier helps Connex to accurately track the sales conversion with the matching user click.

In addition to the two main tasks of click-tracking and conversion-tracking (Google AdWords Help, 2016), the Connex platform can typically provide additional features such as a customised Web Portals for the Best Cars to manage their AM Systems, affiliate commission payment on behalf of Best Cars and sourcing new affiliates and managing them to expand the affiliate network.

### 2.3 Tracking Process

There are two parts to the tracking process: click-tracking (to record a visitor’s mouse-click-action at Globetrotter website) and conversion-tracking (to record an

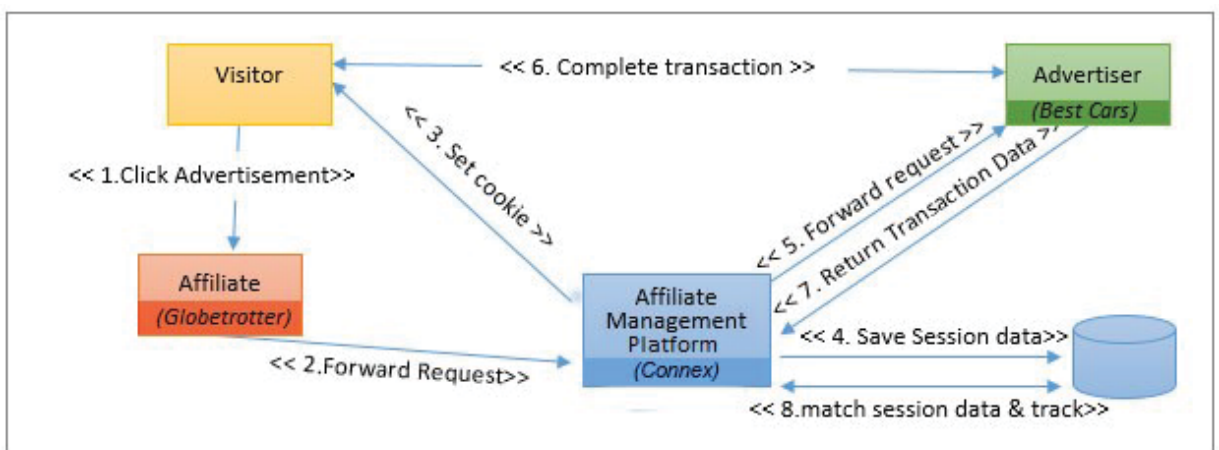


Figure 2: A logical view of the Tracking process

action, such as a reservation that happened at the Best Car's web site). Figure 2 shows the complete tracking process of a specific transaction, which involves the process of matching a conversion-tracking against a corresponding click-tracking. This process allows the AMP to ascertain which affiliate's site generated that traffic, so that the affiliate can be rewarded with the agreed financial gain, in form of a commission payment. If a conversion-tracking does not have a corresponding click-tracking, such transaction would have occurred through a customer typing the URL of the advertiser directly in to the customer's browser or through an organic search or a paid search, but not through an affiliate's web site.

### 2.3.1 Click-tracking

A visitor to the Globetrotter's New Zealand travel blog page clicks on the banner advertisement, which has a URL that points to Connex, instead of pointing to the Best Car's landing page. Connex will place one or more cookies, which are usually known as "tracking cookies", in the visitor's cookies folder of the computer. These cookies contain information that Connex would use for tracking and identification purpose, including a unique Session ID that would identify the visitor again at Best Cars website, on subsequent visits. It will create a Session Cookie with the same Unique ID, which could be used for tracking in case the browser has disabled cookies, which will only last during the current session. After saving information that was passed as URL parameters to a database, Connex will forward the request to the correct product page (landing page) of the advertiser.

### 2.3.2 Conversion-tracking

The visitor usually does not notice that the request to the advertiser's product page has been routed via an AMP, because the request forwarding happens transparently to the customer, at a very high speed. When the site visitor makes a booking, during the confirmation process, the Conversion-Pixel placed on the advertiser's confirmation page will pass the sales data such as session id, transaction id and total price, back to Connex. If the site visitor leaves the advertiser's site without making a booking, the "Conversion-Tracking" will not occur and it will be marked by Connex as a "click" (visit), only. If the site visitor returns to Best Cars website and complete a transaction at a later date, the conversion can be only matched against the first visit through session id and the

affiliate will earn his commission, only if the cookie that was placed during click-tracking is still valid. Even if the site visitor has disabled cookies, the session cookie that was placed during the click-tracking process, will be still active and available if the site visitor completes the booking transaction within the same session, before closing the web browser.

## 2.4 Findings: Risks and Vulnerabilities

An in-depth examination of the processes of a bespoke AMP integration and reconciliation application was undertaken. This application reconciles transactions recorded in the AMP with the back-office databases of the advertiser, using the Web API of the AMP. Such an application is an effective first step in minimising fraudulent activities, as the application filters out any fraudulent sales transactions that do not have a corresponding actual transaction record in the back-office databases of the advertiser. Such filtered-out transaction data include bogus sales transactions that never occurred, which a rogue affiliate might have caused to enter in to the AMP or falsely altered transaction amounts to earn higher commissions, etc. The data from AMP showed all the clicks (website visits) and the conversions (visits converted to bookings). The conversion data included the total revenue earned by the booking and a booking reference number. As this advertiser implemented a “Revenue-share” advertising model, affiliates were paid a commission only for “conversions”. Hence, I was aware that the AMP reported conversions might include genuine conversions as well as possible fraudulent actions that imitated a conversion. If the advertiser implemented a CPC model, I could have seen many more fraudulent activities that would have caused an increase in the amount of clicks (visitor traffic), because the affiliates are paid for each click. It is often easier to manipulate click data than conversion data, hence CPC model is prone to more fraud activity than CPA. Nevertheless, click data has also been examined to uncover any patterns or signs of fraudulent activities. AMP generated web data and the resulting log files were examined, which enabled me to isolate affiliate generated fraudulent web sales data and possible erroneous data caused due to bugs or limitations of technology.

It is suggested that around 1% visitors to an affiliate site actually clicks on an advertisement link (Benediktova & Nevosad, 2008). A rogue affiliate can instead cause, what I call a “*Load-time click*” by adding the trigger-code on “page load” event, instead of “click event” thus mimicking, as if every visitor has clicked a link, or clicked all the advertisement-links in the page. Though the visitor might not have clicked on any advertisement, the visitor would not be aware of the “behind-the-scene” clicks. This fraud is easier to implement on CPC model, which only tracks the clicks. A CPA model needs to use cookies for conversion-tracking purpose, hence, without an “Access-Control-Allow-Origin:[domain name]” header from AMP’s web server, CORS (Cross Origin Resource Sharing) restrictions on cross-site scripting (XSS) prevents Load-time clicking, when the browser and server need to exchange cookies. Therefore, in case of CPA scenarios, “*JSONP*” (*JavaScript Object Notation with Padding*) or an embedded “*iframe*” can be used to achieve a similar result. This fraud can effectively be used, in combination with the “Cookie stuffing” method discussed below. It can insert cookies from diverse advertisers, into a user’s browser, without any visual clue to notify the computer user. These findings have been simulated in the AMP prototype.

Edelman and Brandi (2015) have discussed currently known affiliate frauds in considerable detail, some of which are mentioned below. For the purpose of their paper, they have limited the discussion of technology, by stating that browser “cookies” are the fundamental feature that enables tracking. In this thesis, I have expanded this topic further by describing the technical aspects of cookie tracking and fraud scenarios. I have presented new fraud scenarios that I discovered in the process of developing the integration application and during simulation sessions using the prototype.

*Cookie stuffing* discussed by Edelman & Brandi (2015) is a process, in which an affiliate will place many different cookies belonging to different advertisers (third-party cookies) in the visitor’s computer. If the visitor subsequently visits any of those advertisers’ sites, and makes purchases, the affiliate will earn a commission, without actually having taken part in leading the visitor to that site. The fraud can be combined with “Load time click” fraud, to maximise the commission earnings

by an affiliate. The lifespan of the cookie is an imperative factor, which is often an individual business decision of each advertiser.

An affiliate can do, what I call a “*Conversion hijack*” by monitoring the activity of a direct visitor who did not come through affiliate’s site, but who is about to make a purchase. By triggering a click-pixel just before the purchase, the affiliate places a cookie on visitor’s computer and hijacks the conversion to earn the commission for a sale that would have happened, anyway. Such monitoring can be done using Adware (Edelman & Brandi, 2015) or similar malicious software installed on a visitor’s computer, by an affiliate, which can be described as an external threat. An internal threat, that could cause such monitoring to happen would be when an affiliate gets the help of a staff member of advertiser’s company with sufficient rights, to embed a small code segment on the Web Server, as I uncovered during the implementation of the prototype.

Apart from technology based frauds mentioned above, rogue affiliates can undertake other fraudulent activities that can incur heavy losses using simple deceitful actions. On most e-commerce sites, where a consumer books a service weeks or months ahead of time, such as at tourism related e-commerce sites, an affiliate can book a car or a room himself, a few months ahead and eventually cancel the booking, thus earning a hefty commission. The AMP integration application that I examined for the purpose of this thesis was capable of effectively controlling this category of fraud, by reconciling AMP conversion records with the back-office databases of the advertiser.

From the advertiser’s perspective, some threats originate from external sources such as, by affiliates, site visitors and hackers, while others are internal threats attributed to advertiser’s staff with appropriate security access levels, contractors, IT service providers, etc. Internal threats can be more severe as internal staff can have unrestricted access to, and a comprehensive knowledge of, the IT systems. The most significant risk I established and subsequently tested using the prototype, is what I refer as “*Conversion Stealing*”. It is the process of selecting legitimate transactions from advertiser’s back-office databases, which did not originate through any affiliates, e.g. direct traffic or traffic generated through



search engines, and creating tracking entries on the AMP servers, attributing them to a specific rogue affiliate, who will earn the commission. As many advertisers can have more than half of the on-line sales originating from sources other than affiliate marketing, *conversion stealing* can lead to large losses for an advertiser. This risk can originate internally or externally, though it is much more easier to implement from within the organisation of the advertiser. Even an AMP integration application might not be able to detect this fraud, unless specifically designed to handle this threat, because these illegitimate conversion trackings, in fact refer to legitimate transactions within the advertiser's system.

*Conversion faking* is another new fraud that was uncovered during this investigation using the simulation model, which is discussed further in the next chapter. This fraud is implemented by embedding the conversion pixel of the advertiser, on a *fake* page that is under the control of the rogue affiliate, and to activate the page causing a *Conversion* to be recorded at the Affiliate Network conversion tracking system.

Some of the other types of fraud such as typo-squatting are considered fraud with some advertisers, while it can be legitimate with others. It depends on the contractual agreements as well as different marketing strategies. Typo-squatting is to register and own domain names that are very similar or close to the advertiser's real domain name, with the expectation of capturing the visitor traffic, who were looking for the legitimate advertiser's site. It can be common typos as well as similar sounding names. Such typo-squatting is used to achieve a multitude of fraudulent activities, while an affiliate would capture those visitor traffic, and invisibly re-route to the intended e-commerce site, after a placing a cookie that identifies the affiliate to qualify for the commission. Some advertisers consider it illegal, as they argue that the visitor was intended as a direct visitor, who would have anyway arrived at the e-commerce site without the need of an affiliate recommendation. But other advertisers consider it to be a legitimate act, as a different typo-squatter could have diverted that traffic invisibly to a competitor instead, hence considers it a legitimate promotion.



All the above methods need intermediate to advanced level of technical expertise, and do not imply the affiliate to be a hacker. Most of these activities are carried out by manipulating the existing tracking technology in a “legitimate” way, rather than hacking or breaking in to a system. The tech-savvy hackers and non-techy fraudsters who are in the other two opposite ends of the scale can also cause further losses to an AM practitioner.

## 2.5 Tracking failures

Tracking failures are not fraudulent activities, nevertheless can cause losses to the affiliates losing their rightfully earned commissions. Usually, saving of cookies is allowed by the client’s browser, unless otherwise explicitly disabled by the user.

Even if cookies are disabled, due to the use of Session cookies, which have a life span of the current session, the tracking will function correctly, as long as the user does not close the browser, between the click-tracking process and conversion-tracking process.

The prototype simulations have enabled me to recognise some situations under which tracking will fail:

- If the user clears the browser cache between the click-tracking process and conversion-tracking process.
- If the user uses the “incognito mode” or similar “*private*” browsing feature offered by most browsers, and does not complete the “conversion” in the same browsing session, instead closes the browser, and starts another session of the browser to continue the purchase. But, if the user completes the conversion, then the tracking process will be successful, as session cookie, which has the lifespan of the session duration, will be used.
- If the cookie has expired when the user returns to complete the purchase.
- If the visitor has more than one browser installed on his/her computer, and uses one browser to visit the affiliate’s web site and later uses another browser on the same computer to navigate directly to the advertiser’s website and

makes a purchase. The second browser has no access to the cookie storage of the first browser, as browsers do not share cookies stored, between them, the conversion will not be tracked.

- If the user uses two different computers, one computer to browse the affiliate's website, but uses a different computer to navigate directly to the advertiser's website to make a purchase. Again the conversion will not be tracked, as the cookie was placed in a different computer.

## 2.6 Information security

Though not associated with direct loss of revenue, compromising of information security is another significant vulnerability in an AM network. Further analysis of the AMP log files have led to discovery of numerous threats to information security due to tracking activity. These were further investigated through simulations using the AMP prototype. Due to lack of understanding of technologies that drive an AM system, most AM practitioners are not fully aware to what extent, the privacy of their critical business data can be compromised. When the conversion-tracking pixel is placed on the advertiser's confirmation page, each time a customer completes a transaction, the pixel fires the information back to the AMP, which in turn would match the conversion with a click-tracking. If there is no matching click-tracking found, the commission is not paid to any affiliate, as it would be either a direct sale that did not generate through an affiliate site, or, it could be due to any of the reasons discussed under the "Tracking failure" section. Many advertisers are not aware that the AMP has a running sum of each electronic transaction or booking that has occurred on the advertiser's web site.

If the advertiser chose to use an Analytics service such as Google Analytics or Universal Analytics, or tagging services such as Google Tag Manager, on top of the AMP for pixel tracking, then the advertiser exposes all the electronic business transaction data to Google, as well as to the Affiliate Marketing Platform. This could be in breach of privacy laws in some countries. Hence, Google requests all Universal Analytics users to display a notice to the effect that cookies are used to track specific information about users, which would indemnify Google from privacy infringement accusations. In case of using Google Tag Manager and similar,

that are gaining wider popularity among online marketing practitioners to “manage all in one place”, the information security threat could be further compromised, since Google and the third-party Affiliate Network, both have a record of all the web based sales details and other visitor information. Visitor information indicates what product the visitor was interested in, how long the visitor spent browsing that product, and if the visitor finally purchased the product or not. In case the visit did not convert to a desired outcome, that visitor can be identified as a potential customer at a competitor’s site. Analytics services can then be applied further to use re-marketing strategies to sell this valuable information about potential customers to one’s own competitors.

## 2.7 Summary

Affiliate Marketing is an indispensable marketing tool for e-commerce practitioners around the world, as the advertisers spend their advertising budget only on monetary outcomes. Though it is considered to be “safe”, there are specific inherent risks associated with AM, some are based on vulnerabilities in the cookie-based tracking systems, while others are conventional fraud, that are common to many other marketing models. CPC marketing model is inherently more vulnerable to fraud scenarios, than CPA model. Under specific circumstances, the tracking process can fail, while information security is another important factor that needs to be considered. Affiliate management and cookie tracking activities can be managed in-house by AM practitioners or it can be outsourced to a third-party Affiliate Network. Both options have definite advantages and disadvantages, which have to be considered by a practitioner, before embarking on an AM strategy. A case study was discussed, and the use of a simulation environment to study different fraud scenarios discussed.

In the next chapter, I present the design objectives and implementation details of the invaluable simulation environment AMNSTE, which allowed us to explore the risks and vulnerabilities, and further enabled us to find solutions to mitigate those risks.

## 2.8 Acknowledgements

Chapter 2, in part is a reprint of the material (Exploring Risk and Fraud Scenarios in Affiliate Marketing Technologies from the Advertiser's perspective) as it appears in the proceedings of the Australasian Conference in Information Systems (2015). The thesis author was the primary investigator of this paper.

# Chapter 3

## A simulation and testing environment

This chapter presents the prototype of an Affiliate Marketing Network Simulation Platform. This prototype has been used to investigate different fraud scenarios and has helped uncover new types of risks and vulnerabilities. By simulating some of the known fraud scenarios, I analysed vulnerabilities within an AM platform which can be exploited by affiliates or internal sources. Further, technical solutions to keep business operations safe and mitigate these risks have been implemented. Such an application will help AM practitioners and researchers alike to understand risk environments better and view some of the strategies suggested in defining their policies for online business operations. .

### 3.1 Design Objectives and specifications

This research had a need for an Affiliate Network simulation environment that is technologically identical to the real-world environment that fulfils three needs.

- An environment where frauds known so far can be simulated and tested to allow further examination on the different ways that each specific fraud can be implemented. This enables a comprehensive understanding of the problem at hand. Many frauds can be implemented in different ways, for example: a “load-time click” can be implemented by using an iframe, a JavaScript that runs at the page load event or as any other type of resource request, subject to Cross Origin Resource Sharing (CORS), which will be discussed further. Hence, a simulation environment will greatly benefit in fully understanding the different methods of exploits.
- An environment where new vulnerabilities that could lead to future exploitations can be discovered and tested through simulation.

- Finally, a simulation environment that can be used to develop and test the efficacy and utility of the developed solutions.

This chapter outlines the design and development of a prototype to fulfil the above three needs.

A fully fledged real-world implementations of an Affiliate Marketing Network (AMN) is complex, with Affiliate Networks offering numerous additional features that are not directly related to the tracking process. Such features include recruitment and management of a network of affiliates and offering comprehensive reporting capabilities to affiliates as well as to advertisers. Affiliate Networks also provide a customised portal for each affiliate where the affiliates can find new offers, download HTML code for Click-Pixels and download a collection of banner advertisements to be used in affiliate's webpages. Click-pixels are HTML code segments that are customised for each affiliate with a unique identifier, which need to be embedded in affiliate's web pages, These Click-pixels are attached to an advertisement image which, when clicked by a site visitor, causes to track the affiliate generated traffic activity on a tracking server.

E-commerce sites of Advertisers too have numerous features to offer a satisfying shopping experience to the user, and a range of security implementations including secure on-line payment systems. As we are only interested in the cookie-based tracking process of the AM model, all the non-essential features of each application platform is abstracted, allowing us to concentrate on the core technology behind the tracking process. While abstracting to minimum functional requirements, we have ensured that the AMN solution contains all the necessary features to simulate all risk and fraud scenarios, associated with the tracking process.

Within an AMN solution, three different web applications are required to simulate a real-life AM scenario; an e-commerce site representing an advertiser, a web application providing web services representing an Affiliate Network and multiple instances of affiliate websites representing different affiliates. This allows us to simulate many of the frauds and their effects accurately, such as observe cookie overwriting scenarios, the ability to share a commission between multiple

affiliates, or how multiple redirects are used to avoid detection (Chachra et al., 2015; Snyder & Kanich, 2015; Vacha, Saikat, & Yin, 2013). Each application has to be hosted in a different domain, which influences the ability to enforce cross-domain restrictions, when applications access resources across multiple domains.

The Affiliate website needs a simple user interface with one graphic image representing a banner advertisement of an advertiser, with attached JavaScript and CSS files. These JavaScript and CSS files will be used to simulate frauds that are based on resource requests. A static HTML page will demonstrate frauds that are confined to pure HTML based fraud activities, while a server-side active page is needed to add additional capabilities such as changing the user-agent field dynamically with each web request, to avoid detection (Chachra, 2015). Using multiple Affiliate websites within the AMN solution provides better data samples for analysis, as well as the capability to test “redirect” fraud scenarios, where two or more computers work in unison to commit fraud.

Advertiser’s e-commerce site need a Landing page, with the facility to enter a numeric value and a submit button to post-back a transaction to the server, which mimics an online purchase action of a customer. This page should also save the transaction information and transaction metadata into a database and present a confirmation page to the customer as proof of purchase. This confirmation page will also contain the Conversion pixel, which is a conversion tracking HTML code segment, which would send the conversion data to the tracking server of the Affiliate Network, through a resource request, without any visual clue to the customer.

The Affiliate Network web application needs two separate web service URLs, one for click-tracking and the other for conversion-tracking purposes. This application needs to log information about each click-tracking and conversion-tracking to a database. The application also needs a process to match each conversion-tracking information with an existing click information to determine a successful conversion. The Affiliate Network web application also needs to offer a Web API to the advertiser to allow CRUD (Create, Read, Update and Delete) operations on conversion information, secured by authentication.

Each application requires the capability to store and retrieve information on transactions and metadata of the transactions, in a manner that allows a user to run extensive queries to analyse the historical data to determine fraud and risk scenarios.

### 3.2 Description of the Prototype

The above requirements have been implemented by developing the AMNSTE solution, as described in this section. AMNSTE is designed to represent an Affiliate Marketing Network (AMN) that uses a third-party Affiliate Network platform to manage the AM strategy of an advertiser. Hence the AMN application consists of three different web applications; an e-commerce site of an advertiser, a third-party Affiliate Network, and the web site of an affiliate. Abstracting to minimalistic functional requirements has allowed rapid development of the application with all the necessary functionality and at the same time focus on the fraud discovery tasks without the clutter of unrelated functionality of a real-world specimen of the same applications.

Despite the reduced functionality, all three applications are developed in a layered multitier architecture and modular in design, as found in any real-world application scenario, thus allowing easy separation of functionality into different layers (as shown in Figure 3). Modular representations can be easily customised to suit different implementation requirements and allow updating of modules easily and selectively (Meyer & Webb, 2005). This also facilitates the testing of feasibility of applying different solutions, such as integration applications for reconciling transactions, etc., in the same manner as a real-world application. Among the three applications, during the testing phase of frauds and risks, most of the modifications and updates usually occur in the affiliate's website, as it is the source of most of fraud activities. Some fraud activities originate in the advertiser's e-commerce site, while others can originate at Affiliate Network Platform. In contrary, during the solution development phase most updates take place in the Affiliate Network Platform and in the advertiser's e-commerce site, where such solutions are needed. The modular component design allows adding of



functionality to any of these applications, simply by updating individual modules independent of other modules, without breaking any code or program execution.

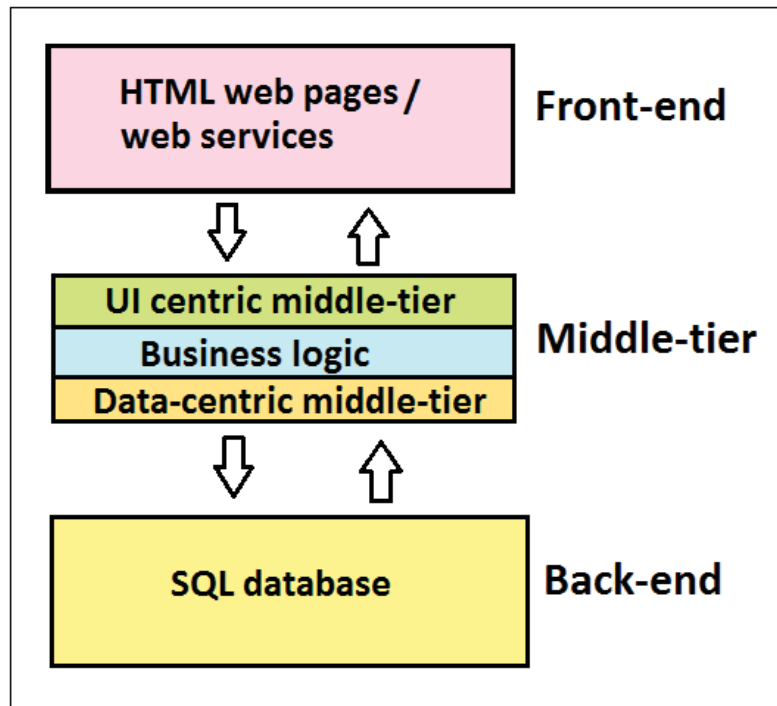


Figure 3: Layered n-tier architecture

Following the layered architecture as shown in Figure 3, the middle-tier is further divided into UI centric and data-centric layers. Within the affiliate website each fraudulent action is represented with a separate UI, for demonstration purpose. While the business objects reside in the business logic layer, data-centric middle tier populates the business objects with data from the underlying data sources. Through the separation of the data-centric middle-tier, the application can be connected to any data source, of user's choice. This data-centric middle-tier is implemented within each of the three applications by a single Data Access Layer (DAL) class each, named AdvDAL (advertiser), ConnexDAL (Affiliate Network), and AffDAL (affiliate). Connection strings to different data sources such as Microsoft SQL, Oracle, MySQL, portable databases etc. are stored in the *web.config* file and an entry for the default connection is stored under *appSettings* section of the *web.config* file. When Business Objects are populated or persisted back to the data store, the relevant method calls the *GetConnection()* method within the DAL, which creates a connection to the database of user's choice, stored as the default database. If different data stores use different field names or table structures to

store the data, an additional mapping class is used to map the data store's data fields to the appropriate properties of the business objects. The separation of DAL class enables connectivity to multitude of data sources such as databases, flat files, etc., for data storage and retrieval of the transaction data, transaction log files and error log files. A user can choose a data store simply by selecting the choice from a list. Adding of any new data stores only need a new entry of a connection string in the *web.config* file and a connection/data mapping in the DAL class, without any impact or change in the rest of the middle-tier.

All three web applications are designed to store transaction data and metadata relating to those transactions, such as the session identifier, request method, user-agent, referrer etc., within each corresponding database. This has allowed us to run queries against the databases and analyse the transaction data and metadata to validate known risks, -further described in chapter 4.

The AMNSTE is usually deployed in a compiled and hosted state on web servers. In such state, all transaction and metadata are stored in a database and the fraud and risk detection is carried out primarily by executing queries against the SQL database and analysing the resulting data. In addition to the compiled state, as we designed the AMN simulator within Microsoft .NET development environment, AMNSTE allow researchers with technical expertise to run simulations within a development environment. This "Live" environment with advanced debugging capabilities will allow researchers additional control of program flow and higher manipulation capabilities, without relying solely on the data that will be stored in the database. Though other software development platforms also have similar capabilities to varying degrees, Microsoft .NET Development environment was chosen, as all three applications can run simultaneously within one solution, which allows us advanced debugging and control, not only within each application, but also, allows us to watch, monitor and control the communication and interactions between the three applications, within a "Live" environment.

### 3.3 System Architecture

AMNSTE is implemented using Microsoft Development Environment, hosting all three applications within one solution, for easy testing and debugging purposes. It

can be equally well implemented in a LAMP (Linux/Apache/MySQL/PHP) or a heterogeneous environment as in case with real-world AMNs. To enable the closest real-world environment, in this solution four virtual machines (VM) running on Windows Server 2012 operating system were each configured to be in a separate network domain, with local routing and Domain Name resolution. It is important to host the applications within different network domains to test CORS (Cross Origin Resource Sharing) restrictions, XSS (cross-site-scripting) features and X-Frame-Options HTTP header in iframes, as well as any future cross-domain restrictions. Each VM implements Microsoft Internet Information Server (IIS) as the web server and Microsoft SQL Server 2012 as a back-end database server, which store the transaction and tracking data relevant to each web application. Two VMs are used to host affiliate websites, while one IIS server on one of the VMs that is hosting affiliate websites, is configured for a shared hosting environment with medium trust setting, which enables us to run as many multiple instances of affiliate websites as needed, and test the fraud capabilities, also in shared hosting environments. The sequence diagram below (Fig. 4.) depicts the interactions between the three web applications that make up the AMNSTE.

### 3.3.1 Affiliate Network platform

In most cases, the advertisers use a third-party Affiliate Network, which is responsible for the tracking of affiliate transaction data. The tracking process usually captures Affiliate identifier, date and time of the transaction, transaction identifier, and the transaction amount. This Affiliate Network platform follows a service oriented architecture (SOA) and is modular in design (Fig. 5). The main purpose of the Affiliate Network platform is to track click actions and conversions, which it fulfils by offering a set of web services. The web services are implemented using generic handlers (.ashx) and do not have a user interface. In a real-world scenario, Affiliate Network platforms usually have additional features such as an affiliate portal where each affiliate can securely log-in and see detailed information about the click traffic generated by the individual affiliate's website and the details of the successful conversions. Affiliate Networks also have a comprehensive Web API that allows advertisers to connect securely to retrieve and edit tracking data. But for the intended purpose AMNSTE concentrates on the core functionality by

offering a web services URL each, for click tracking, conversion tracking and a Web API for CRUD operations of the tracking data. The tracking data along with associated metadata such as the *user-agent* and *referrer* information are saved in a Microsoft SQL database.

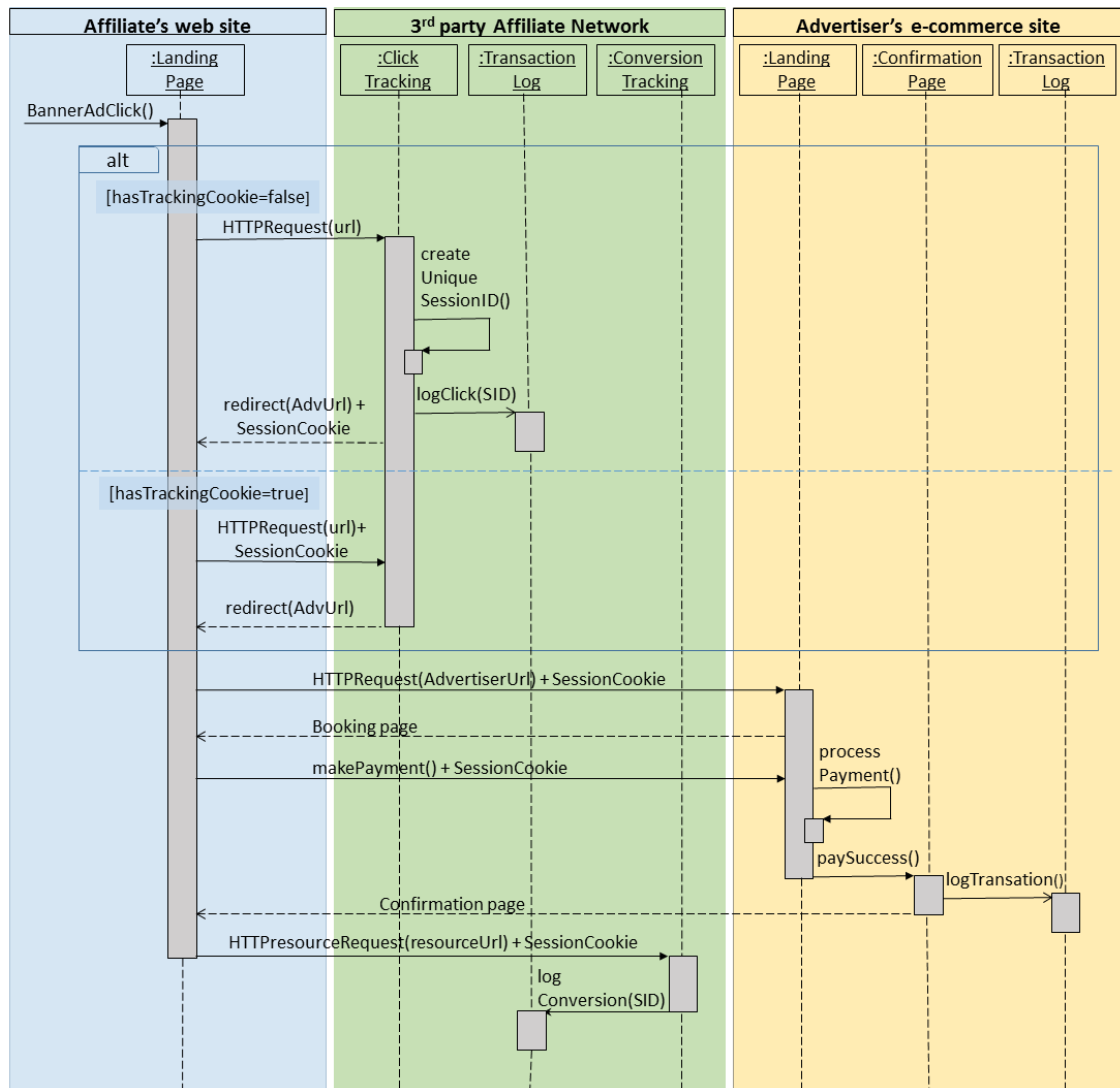


Figure 4: Sequence diagram of the tracking process

As shown in Figure 4. when a visitor clicks on an advertisement at an affiliate's website, the "Click-tracking" code in the HTML page causes the browser to send an HTTP request to the click tracking URL "http://<Affiliate Network domain>/click.ashx?a=3&f=5", which has two parameters, Advertiser identifier (a=3) and Affiliate identifier (f=5). It can contain additional parameters for additional features such as Campaign identifier, Revenue model (CPC, CPA, Revenue share, etc.). The *Click.ashx* generic handler class (Fig. 5) handles the incoming HTTP request, *logClick()* method create an instance of a *ClickTracker*

class and executes the *TrackThisClick()* method, which extracts and checks the validity of the URL parameters that were passed, and the availability of a cookie that has been set by the Affiliate Network, previously. If the visitor's browser does not forward a cookie with the HTTP request, it denotes a first time visitor, in which case a new cookie will be created with a new and unique session identifier and returned to the browser.

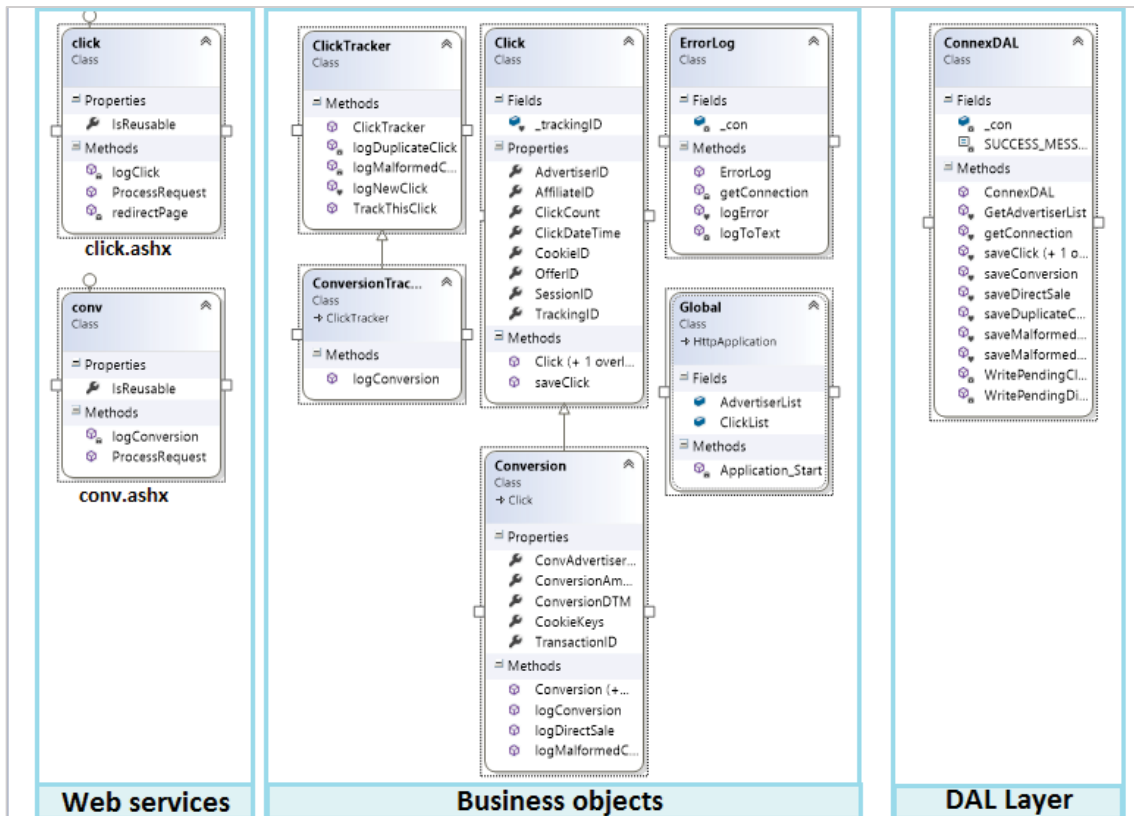


Figure 5. Class diagram of Affiliate Network application

The cookie contains information such as the session identifier, which uniquely identifies a visitor, affiliate identifier, advertiser identifier and the date of the visit. It can store additional data fields to fulfil additional business rules as per requirements, which are passed as HTTP request parameters in the URL. If the browser submits a cookie that has been previously set by the Affiliate Network, which contains same advertiser and affiliate identifiers, it denotes that this visitor has previously visited the same advertiser, via a page of the same affiliate, but has not completed a sale, as otherwise the cookie would have been deleted after a successful sales conversion. In this case, since there is no need to place a new cookie, the existing cookie's validity period will be renewed, so that the affiliate

has a chance of earning a commission over a longer period of time. If the cookie that was forwarded by the browser contains a different advertiser or affiliate identifier than the identifiers contained in the current URL parameters, this would mean the visitor has previously clicked an advertisement of a different advertiser or clicked an advertisement that appeared at a different affiliate's page. In either case, the cookie will either be overwritten with the new advertiser and affiliate identifiers, or, the current advertiser and affiliate identifiers will be added to the existing list of identifiers in the cookie, depending on the business rules implemented by the advertiser.

After creating a new cookie or editing an existing cookie by the *ClickTracker* object residing in the *Business Logic* layer (in the middle tier of the AMNSTE), the somewhat more processor intensive task of persisting the click-tracking data is passed on to the *Data-centric* layer of the middle tier. In this manner any latency caused by database activity is prevented, and the server redirects the client browser to the advertiser's e-commerce site by looking up the corresponding advertiser's URL against the advertiser identifier within the Global Dictionary *AdvertiserList*. Accordingly, *logNewClick()* method saves the click-tracking information as a new and first-time click in to the database, or in case of an existing cookie *logDuplicateClick()* method will cause the database to first retrieve the existing click-tracking data, and increment the *click count* field and update the *Last click date*. Though in a simulation environment with low activity, this task does not cause any latency, with a high-traffic production environment in mind, the two tasks of retrieving the existing record and updating *click count* and *Last click date* is been run on the database server within one stored procedure. This stored procedure returns a value to indicate the success status to *logDuplicateClick()* method. As the *logDuplicateClick()* method has to wait for the status code to return, it is run as an asynchronous task. In case a status code denoting a failure is returned, retry efforts and saving to a log file for future update or for manual intervention are carried out as per user-defined business rules.

As discussed earlier, with the layered architecture, we are able to use clustering and load-balancing techniques for further improvement of latency, as and when required in a high volume production environment. This can be carried out easily

by only updating the Data-centric (*ConnexDAL*) layer, without effecting any other part or modules of the software. If any of the URL parameters of the HTTP request is invalid, such as in case of an invalid advertiser or affiliate identifier, the click is handled by the *logMalformedClick()* method of the *ClickTracker* object, which passes on the task of saving the data in the *MalformedClicks* table, to the *ConnexDAL* object. Records in *MalformedClicks* table can be manually inspected to detect configuration problems in affiliate webpages or as possible fraud attempts, as we discuss further below.

When any visitor to the advertiser's e-commerce site performs an action that earns a commission or a fee for the affiliate (e.g. purchased a product or service or signed up for a contract, etc.) the conversion-tracking code on the advertiser's confirmation page, upon rendering the page on client's browser, sends a HTTP resource request to the conversion tracking web services URL on Affiliate Network application's web server, similar to:

"http://<Affiliate Network domain>/conv.ashx?a=3&t=12345&p=250.00"

The parameters embedded in the URL identifies the Advertiser identifier (a=3), Transaction identifier (t=12345) & the Total Price (p=250.00). The browser also sends any cookies in its possession that have been previously placed by the Affiliate Network, together with the HTTP request. As in click-tracking process described above, during the conversion-tracking process, absence of an accompanying cookie denotes, that this particular sale has originated from a direct customer, who did not arrive at advertiser's e-commerce site through any affiliate's website. This denotes an "Organic Traffic" paid search, third-party links or the visitor typing the URL directly in to the browser. Organic searches are search results returned by a search engine such as Google, Bing or Yahoo, without using paid advertising, instead purely on the merit of the page ranking.

An instance of a *ConversionTracker* is created and entrusted to process this instance of the conversion. A new *Conversion* object is created with the data extracted from URL parameters, and it is validated for data accuracy. If the parameters passed by the URL cannot be validated, it represents a *conversion* that cannot be reconciled, hence will be passed to the *logMalformedConversion()*



method for manual examination. Again, the *ConnexDAL* in the Data-centric middle tier, is entrusted the somewhat processor intensive task of saving the transaction to the database. As there is no user interface for the Affiliate Network application, and no need to display the result of saving *Conversion* details to the database, *ConversionTracker* does not need to wait to get a result code from the underlying *ConnexDAL* or database. To minimise the latency within the application, *ConnexDAL* uses a stored procedure in the database, to first find the *Click-tracking* record that relates to the current *Conversion* and save the *Conversion* to the database, merging with the *click-tracking* information.

Each valid *Conversion* that has an accompanying cookie from the Affiliate Network, has been *Click-tracked* before, therefore, is expected to have a valid click-record, as none of the direct-sales will have an accompanying cookie. Hence, if no click record is found, the stored procedure in the database, alternatively, save the record in the *ConversionQueue* table for manual intervention. Such record may depict a failed attempt to save a click tracking, which therefore is still awaiting manual intervention, as discussed previously in the click-tracking process, or it could represent a fraud scenario of a cookie manipulation. Both of the above cases will warrant manual intervention. At this stage the cookie can be deleted by the Affiliate Network, which is done by changing the validity period of the cookie to a past date. Doing so, restricts the affiliate from earning any further commissions from the same visitor, depending on the business rules of the individual advertiser. Such business decisions can be implemented through a “settings” page of the application.

A multitude of information can be gathered by executing SQL queries against the database. Click-tracking information can be identified with corresponding Conversion-tracking information using a matching session identifier. Numerous fraud or risk related behaviour can be identified through data analysis which are discussed in chapter 4.

### 3.3.2 Advertiser’s e-commerce site

The Advertiser’s e-commerce site has two HTML pages, a Landing page (depicted as “Booking page” in Figure 4) and a confirmation page. The first page is an HTML



page designated as the Landing page, which contains a text box named “Total Price” and a submit button. This simulates the buying of a product for a specific price, by a visitor to the e-commerce site. The sequence diagram in Figure 4 shows a combination of actions such as the selection and adding of products in to the shopping cart and payment confirmation, summarised into a single action as *makePayment()*. In a real-world e-commerce site this involves browsing through numerous pages or going through numerous steps using AJAX in an asynchronous environment, which is abstracted in order to concentrate on our primary requirements. The same two HTML pages described above have been also implemented with same functionality as two server-side ASPX pages, to enable us to test different solutions to discovered vulnerabilities during the solution testing phase.

The Conversion-tracking code has been added to the confirmation page as an HTTP resource request, as the source of a JavaScript file and within an iframe (in case the client browser does not support JavaScript) for high reliability (Snyder & Kanich, 2015). Alternatively, Conversion-tracking can be implemented as a server-side process, which is executed after the payment confirmation, before serving the confirmation page. Client-side implementation as HTML code is preferable for third-party Affiliate Networks, which makes both tracking implementations, click-tracking and conversion-tracking, a client-side technology, which allows web sites to add the HTML tracking code to their pages, without advanced technical expertise.

### 3.3.3 Affiliate’s Website

The affiliate’s web site has a HTML based web page with one graphic image representing a banner advertisement. The image is a hyperlink to a tracking URL on Affiliate Network. The URL contains parameters that include the affiliate identifier and the advertiser identifier. In a real-world application, additional information that need to be tracked, such as campaign identifier, banner identifier, etc. can be passed as parameters, enabling more functionality. This page has an attached JavaScript file, enabling us to use JavaScript code to test different frauds such as “Load-time click”. Most of the fraud tests will be carried out on the affiliate’s website, though, as has been discussed in the previous chapter, some

frauds can originate external to the affiliate's website. Though in real-world scenarios, most affiliate websites will be pure HTML pages, we have also included a Microsoft ASPX Web Form with same features and functionality as above, which allows us to dynamically change parameters such as *user-agent* string, which also enables to automate numerous fraud scenarios and store visitor data in a database for post-analysis during simulations.

Though the primary objective of the application is to test for risk and fraud scenarios, numerous customer behavioural information too can be identified through data analysis. For instance, SQL queries can be executed against the back-end database to extract information to show how many times a visitor visited the affiliate's site before eventually making a purchase through the number of click-trackings with the same Session identifier. The referrer field of the HTTP request allows us to discover: what web page was used by the user before arriving at the affiliate's website, their IP address which allows us to extract the geographic location, the user-agent field for "browser fingerprinting", etc.

Frauds based on application code and scripting can be implemented within automated processes as a scheduled tasks on the affiliate website. It also allows us to replicate in large volume, all the frauds that are been tested, thereby creating large datasets. Another approach is to create an HTML page for each known fraud, thus enabling us to demonstrate each fraud individually.

### 3.4 Using the prototype system

The AMN simulator can be used in two different environments, either in a development or a production environment. When it is compiled and hosted in a *production environment*, it becomes rather user-friendly, though in such case, fraud and risk detection is carried out primarily through data analysis, by executing queries against the SQL database. When the AMN simulator is used Microsoft .NET *development environment*, it allows a researcher with technical expertise to run a "Live" environment with debugging capabilities, which allows us to monitor "Live", processes within a single application, as well as its interaction with the other applications. The AMNSTE has been used firstly to study fraud and risk scenarios, and then to develop solutions to address those vulnerabilities.

### 3.4.1 Using AMNSTE to discover frauds and risks

AMNSTE can be used to discover a multitude of methods to implement some of the known frauds. Such knowledge becomes useful to develop solutions that can address all the different variation and implementations of the same fraud. AMNSTE also helps us to discover new vulnerabilities that can be utilised by rogue affiliate in future.

For the purpose of demonstration, a new static HTML page or a dynamic ASPX page was created, depending on the requirement, to demonstrate each fraud that needed to be tested. The result of that fraud can be then observed by querying the databases of the Affiliate Network system, or using debugging techniques, if the tests are being run on a “Live” environment. For example, *Cookie stuffing* is a fraud that inserts one or more tracking cookies to a visitor’s browser, when a visitor arrives at an affiliate’s website, without the visitor’s consent and without the visitor having clicked on any advertisements. Doing so, identifies the rogue affiliate as the source of the visitor traffic, to all the e-commerce sites, whose cookies were placed in visitor’s browser. Any subsequent purchase by the visitor at the any of advertiser’s e-commerce sites will earn a commission for the unscrupulous affiliate. Though many previous researches (Edelman & Brandi, 2015; Chachra et al., 2015; Snyder & Kanich, 2015) consider that the overwriting of an existing cookie, which has been placed legitimately by a different affiliate previously, to be one of the most serious offenses caused by cookie stuffing, our research and further simulations show that the cookie overwriting depends on the business rules defined by each advertiser.

The advertiser can choose to set up an advertising model, where “last person gets all”, under which the last affiliate gets the full commission amount or where “Share Commission” basis is employed, under which all or multiple affiliates can share the commission either equally or proportionately, depending on the business rules. For example, when a visitor who already has a tracking cookie from one affiliate, which has been placed during a previous visit, clicks an advertisement again, of the same advertiser, displayed by a different affiliate, or if an unscrupulous affiliate causes a cookie stuffing fraud, the click-tracking server will place the cookie as follows; if “last person gets all” rule is implemented, then the previous cookie will

be overwritten and the second affiliates gets the commission. If “Share the commission” rule is in place, then the last affiliate’s identifier is added to the affiliates list within the cookie. When the visitor then makes a purchase at the advertiser’s e-commerce site, all affiliates will share the commission. If the commission is shared equally or proportionately is defined by another business rule. Usually, after each successful business outcome, the cookie will be deleted, preventing further commission payment to the hitherto known affiliates. Such features are offered to the advertisers by the affiliate marketing networks, to implement advertiser’s own choices as individual business rules. Those capabilities are unique selling points (USP’s) of different Affiliate Networks.

Cookie stuffing fraud can be implemented in multiple ways. Using invisible iframes within the landing page of the affiliate, where source property of the iframe is set to the click tracking URL, results in the Affiliate Network sending a cookie to the browser and identifying the affiliate as the source of tracking. We also replicated this fraud successfully using an HTML image tag, where the source property is set to the tracking URL. Further, a dishonest affiliate can maximise his bounty by “stuffing” an array of cookies from diverse advertisers in to the visitor’s browser, hoping the visitor might go to some of those advertiser’s sites, earning him commission at each of the advertiser sites.

Cookie stuffing can be combined with “Load-time-click” fraud, which has been discussed in chapter 2. This can be done by adding the trigger-code on “page load” event instead of “on click” event, to send HTTP GET requests to multiple number of click-tracking URL’s. Our simulation shows that this is possible in case of CPC advertising models, where only clicks are counted, but not for retrieving a cookie to be placed in the browser. CORS (Cross Origin Resource Sharing) restrictions on cross-site-scripting (XSS) prevents the browser from placing a cookie, without an “Access-Control-Allow-Origin:[domain name]” header from the Affiliate Network server. Interestingly, the simulation shows that the Affiliate Network server does send the response with the cookie, but the restriction is enforced at client-side browser, hence it can still be used for CPC.

Instead of using JavaScript code for DOM manipulation, HTML code to retrieve a “non-existent” JavaScript file, given the click-tracking URL as the source, accomplished the same goal successfully in our simulations. Similarly, it was possible to replicate the above fraud equally well using attached stylesheets.

#### 3.4.2 Using AMNSTE to find solutions and test efficacy & utility

When developing and testing solutions to mitigate discovered fraud and risk scenarios, most of the code updates are carried out in the Affiliate Network application and advertiser’s e-commerce application. The modular design of these application allows for easy and quick updated for specific modules, without effecting the rest of the software solution.

AMNSTE helped us discover and confirm through simulation, the scenarios that were discussed above, which in turn allowed us to look for solutions that would address those scenarios, individually or collectively. We found that number of steps can be used to overcome and control the above scenario. Firstly, at the time of sign up, the affiliates can be required to submit the URL’s of the pages, where the advertisement links will be used. Subsequently, on each click tracking instance, “referrer” field In the HTTP header of each click should refer to one of the given URLs. While this verification can be undertaken at each click tracking instance, a web crawler can be used as a scheduled task to crawl each URL discovered during click tracking, to identify any illegitimate code in HTML, JavaScript and CSS files. Any cookie-set commands executed at loading of the page during crawling would be a sure sign of cookie stuffing. A modern browser with a crawler as a plug-in, as used by Chachra et al. (2015) can be used against more technically advanced affiliates. To prevent cooke stuffing using iframe method,

### 3.5 Conclusion and Future Research

AMNSTE contributes to the practical need of a simulation environment for the researchers and industry stakeholders. With the help of this application further research can be carried out to find out solutions to known frauds and risks, and to detect further vulnerabilities that can be exploited by fraudulent affiliates and discover preventative solutions, pro-actively.

Currently the historical tracking data of the AM strategy belonging to a New Zealand tourism operator is being analysed, where services are booked by customers up to many months ahead. The AMNSTE tool has helped us to simulate numerous fraud activities that were discovered thus far. The prototype is even more valuable as a tool, to design and test solutions to control the risks that have been identified. The results of the findings and solutions proposed will be published in the future.

### 3.6 Summary

An AM simulation environment is vital for AM practitioners and researchers alike to study risks and vulnerabilities that are common in the AM ecosystem and to develop and test solutions. AMNSTE comprises of three different Web Applications, an e-commerce site of an advertiser, an affiliate's website and a third-party Affiliate Network Application. All three applications are reduced to functional requirements to create a technologically identical environment as in real-world, while keeping it manageable and focused on the requirements for testing purposes. While AMNSTE can be implemented in a heterogeneous environment as in real-world, this solution was developed using Microsoft .NET Development Environment and each application hosted on a separate virtual machine (VM) system. The modular design of the system allows frequent changes to components easily without effecting other parts of the program flow. Such updates were undertaken to add specific functionality to simulate newer frauds and risks that were discovered during this research. Further updates were needed to add additional functionality to control newly discovered risks, as solutions evolved.

In the next chapter, I present the results of using AMNSTE to analyse the dataset of an AM effort of an AM practitioner that is a car rental company in New Zealand.

### 3.7 Acknowledgements

Chapter 3, in part, is the material that was submitted for the IEEE 14<sup>th</sup> International Conference on Privacy, Security and Trust. The thesis author is the primary investigator of this paper.

# Chapter 4

## Analysing the Data

Best cars (pseudonym) mentioned in our case study in the previous chapter, uses Connex (pseudonym), an external third-party Affiliate Network. Best cars manages the new sign-ups of affiliates as well as monthly calculation and payment of commissions to affiliates, as per the click- and conversion-tracking data supplied by Connex.

### 4.1 Use of two Affiliate Networks

In March 2014, Best Cars changed their affiliate network. All the conversion-tracking data for the period from August 2013 to March 2014 were available for my study. Best Cars considers this period under study, reflects the best period in the company's affiliate marketing strategy, with two strong players providing the bulk of the visitor traffic and the successful conversions. Investigation of the tracking and transaction data for the given period of eight months offered interesting insights. The dataset contained 21,337 online transactions. Of these 17,417 records (82%) were removed from study, since these were generated by organic searchers or direct customers. These records were denoted by an empty affiliate identifier field and an empty cookie id field.

Additionally a second dataset consisting of all conversions that were considered by the Affiliate Network to be successful, legitimate and affiliate generated over a much longer period of forty seven months, from April 2010 to February 2014 was investigated. The second dataset contained 15,221 records. The first dataset allowed examination of fraudulent attempts across all the conversion activity, while the second dataset allowed further scrutiny of those conversions which were considered legitimate. Both datasets were imported into a Microsoft SQL Server 2012 relational database management system and analysed in combination and individually.



## 4.2 Tracking failures

Out of all the conversions in the first dataset, 82% were reported to be either organic searches or direct customers, who typed the advertiser's e-commerce site's URL in to the address bar of the browser themselves. Organic searches are search results returned by a search engine such as Google, Bing or Yahoo, without using paid advertising, instead purely on the merit of the page ranking. As they did not originate at any affiliate website, no affiliate commission was paid for the said traffic. But we observed that 2.5% of the above transactions did accompany a cookie of the Affiliate Network at the time of the booking, which indicated that those customers have been click-tracked from an affiliate website previously, but Affiliate Network has not been able to identify the affiliate who generated the traffic. This is likely due to a malformed URL, which is composed of affiliate's tracking information appended to the address of the advertiser's website. The correct URL is provided by the advertiser at the time of the affiliate sign-up, which eliminates such malformed URL errors. Affiliate Networks offer a customised portal for each affiliate to sign in securely and download a choice of banner advertisements and the URL that is customised with the affiliate identifier that is specific to that affiliate. Any URL errors are generally easily identified at the time of website setup. A test page usually allows the affiliate to click the newly embedded banner and monitor the successful click-tracking event on the Affiliate Network platform. Hence, the 2.5% visitor traffic that was identified as direct traffic, but with an accompanying browser cookie, can be interpreted as a possible failure of an attempted fraud by an affiliate, which we could not ascertain as the Affiliate Network has in the meantime, ceased to exist.

## 4.3 Duplicate Cookies

Usually, if an Affiliate Network only tracks the last affiliate who generated the visitor traffic, then at the time of completing a sale, the Affiliate Network causes the cookie in the visitor's browser to expire, thus duplicate cookies in conversion records, are not legitimate. The common practise is, as it is currently configured for this advertiser, the click-tracking cookie is deleted after the completion of a sale. If that same visitor arrives at the e-commerce site directly in future, without the tracking cookie in the browser, this visitor will be considered a direct visitor



and no commission will be paid to the affiliate. But if that same visitor arrives at the e-commerce site, each time via the affiliate web site, then, each time, a new tracking cookie will be placed at click-time, and the affiliate will earn a commission. Though this is the common practise, some Affiliate Networks offer the advertiser the ability to decide on such policy decisions as defined business rules, which can be initially set up by the advertiser. Overall, 3.8% attempts of all records of which 0.4% attempts of all organic records had duplicate browser cookies.

#### 4.4 Duplicate IP addresses

Further, 20% of transactions used duplicate IP addresses with one IP address being used 33 times. While multiple use of the same cookie or the same IP address within a short period may indicate possible fraudulent scenarios, single occurrence of the same does not necessarily indicate a genuine transaction either. More careful and tech savvy fraudsters would avoid detection by using distributed proxy servers or botnets (Kanich, et al., 2008) to avoid duplicate IP addresses and use “private mode” browsing, such as “incognito window” in Chrome browser or “InPrivate window” in Microsoft Internet Explorer(IE) to avoid duplicate cookies.

The tracking record of each booking transaction contained information such as date and time, affiliate identifier, web browser’s cookie identifier, client computer’s IP address, page referrer, total sum of transaction, transaction identifier, etc. The information have been tracked and saved at the time of the transaction. A browser cookie has been placed in the client computer when the visitor first clicked on an advertisement, and that cookie has been sent back to the tracking server each time the client made a web request. This has allowed the tracking server to identify the specific computer throughout all its transactions with the tracking server that followed.

While the geographic details of the visitor can be accurately identified using the IP address, it cannot be considered a unique identifier of a specific user computer, as a router can assume the publicly visible IP address and share it with numerous client computers connected to the router using Network Address Translation (NAT) routing protocol. But during short periods of time, in conjunction with the “user-agent” identifier, it can be considered to be unique to a specific user

computer. The tracking records above did not contain the “user-agent” identifier; hence we used the IP address only for geo-referencing. Page referrer field indicates the address of the page the user was viewing, before arriving at the target webpage, if the user followed a link to arrive at the target webpage.

All affiliate fraud can be divided into two categories, based on transaction authenticity. One group of frauds are based on *genuine transactions* that have already taken place, which are manipulated by a rogue affiliate to claim a commission. The other group of frauds are *bogus transactions* that have never occurred, but where the rogue affiliate has been successful in creating *Click-* and *Conversion-tracking* records, with bogus data within the Affiliate Network. Advertisers who do not have the technical capability to reconcile *conversion* data provided by the Affiliate Network with the back-end transaction data stores of the advertiser, are vulnerable to this category of fraud.

After carrying out the above transaction data analyses, next major step that was carried out in fraud detection, was the use of a transaction reconciliation application. An integration application was developed which can be utilised to reconcile transactions between a third-party Affiliate Network and an advertiser’s transaction processing data stores. Using this application to reconcile the two data sets was found to be a very efficient way to separate *genuine transactions* from *bogus transactions*. This solution has been proposed in the recommendations section of the final chapter of this thesis.

Within this reconciliation process, all conversions in the tracking database that did not have a corresponding transaction record within the transaction database of the advertiser were filtered out as illegal conversions. How the records are matched is discussed later in this chapter. All the conversions that have a corresponding transaction record within the transaction database are considered “pseudo-legitimate” as they appear to be legitimate transactions, until we find evidence to the contrary. Most of the frauds that we have thus far discussed, such as *Load-time clicks*, *Cookie Stuffing*, *Conversion hijacking*, *Conversion stealing*, fall into the *genuine transaction* category, and deal with real and legitimate transactions that take place at the e-commerce site, where unscrupulous affiliates claim a

commission, without having promoted that business. Nevertheless, the transaction has occurred and the advertiser has in fact earned revenue through that transaction, hence the real loss to the advertiser due to one of the above frauds amounts to losing a part of the profit to a fraudster.

The next fraud detected has been named “*Conversion faking*”, which falls in to “*bogus transactions*” category, is in contrary far more damaging to an advertiser, than the above frauds. The tracking data analysis has revealed that some unscrupulous affiliates have managed to add *Conversion-tracking* records and matching *click-tracking records* to the Affiliate Network database, of bogus transactions, that have never occurred, claiming commissions for transactions that never occurred. There are numerous ways to enter such bogus entries in to the tracking system, including Conversion faking, which is to “fire” the conversion-pixel from a bogus website under an affiliate’s control or doing so programmatically by conversion hijacking or conversion stealing. Hence, during this research, we recognised that a reconcile application is a quick, simple and effective solution to filter out a large number of fraudulent conversions. The advertiser in this study has not had the capability to check the authenticity of each conversion tracking information that was recorded at the Affiliate Network, while the Affiliate Network could have easily detected these frauds by comparing the *referrer* field of the visitor browser’s HTTP request with the legitimate *referrer*, which is the URL of the *payment* page of the advertiser. In those records that were inspected, the bogus transactions had either an empty *referrer* field or a “wrong” *referrer* field. The “wrong” *referrer* field exposed further unscrupulous activity, which are not discussed here, as they do not fall within affiliate fraud category.

During the reconcile process, while matching a conversion-tracking record with a transaction record within the back-end transaction data store of the advertiser, comparing only the transaction identifier is insufficient. The transaction identifier and the total sum of the transaction and date and time are important fields for us to detect this fraud. Any record that did not match the combined fields of transaction identifier, transaction date and the total sum represented a clear fraudulent record, which was rejected. During this process, we filtered 12% of the records which did not have a matching transaction record in the database as

obvious fraud transactions. Unscrupulous affiliates who do not expect that affiliate transactions are usually reconciled, have assumed a random transaction identifier to enter bogus entries in to the tracking system.

Those tracking records that have a matching transaction record in the transaction database are considered pseudo-legitimate at this stage of the reconciliation process, as they appear to be legitimate transactions. These pseudo-legitimate transactions need to be further investigated for more refined fraud than those using randomly generated data or combination of data fields. Each of these pseudo-legitimate conversion records on the Affiliate Network's database refers to a genuine transaction that has occurred. Though an affiliate can guess the next transaction identifier in a sequence, after carrying out one legitimate transaction, we still can detect fraudulent attempts by comparing a combination of data fields. For instance comparing the transaction identifier and the timestamp of the conversion, and to further strengthen the comparison, including other data such as the total price of the conversion in to the comparison will prevent an affiliate from falsely claiming the commission against a genuine transaction that has happened. Even in a rare event of guessing all the three data items correctly, the transaction has to refer to a one that was generated through an organic search; else the transaction would already have been claimed by a different legitimate affiliate. Hence, monitoring duplicate claims as above is an important part of the reconcile process, that can raise a flag when it occurs that allows the advertiser to blacklist the rogue affiliate, without risking further fraud attempts.

In the process of further examining pseudo-legitimate transactions, examining the referrer field is the next most important step. The unique and legitimate conversion-pixel is implanted in the transaction confirmation page on Advertiser's website. Hence, the referrer should always be the page before that, which is usually the payment form. Any other referrer field denotes a fraud activity, usually caused by a rogue affiliate activating the pixel code from a different URL belonging to the affiliate or doing so programmatically.

## 4.5 Summary

This research had the privilege of having access to an affiliate tracking dataset, which allowed me to ascertain and quantify prevalence of fraud scenarios within an AM effort. The results show that this practitioner, being a tourism related operator, who is almost entirely dependent on online sales, generated 20.5% of the sales through AM, though 2.5% may not have been correctly attributed to an affiliate. 12% of conversion records were clear fraudulent activities, as those records used bogus transaction identifiers. The actual number of attempted frauds are unknown to us, as this record set comprised of records that were deemed to be authentic after the Affiliate Network's scrutiny. Many other signs of possible fraud scenarios were observed. The most significant vulnerability that we discovered was a hypothetical possibility that has been named *conversion claiming*. The most significant solution that we found to counter many of the frauds that were discovered was the implementation of an integration application to reconcile tracking records from the Affiliate Network platform, with the back-end transaction data stores of the advertiser.

## 4.6 Acknowledgements

Chapter 4, in part, is material in the paper (Revenue Fraud in e-commerce platforms: Challenges and Solutions for Affiliate Marketing) as it appears as a chapter of the book "Cyber Security and Policy", Auckland, New Zealand: Massey University Press 2016. The thesis author was the primary investigator of this paper.

# Chapter 5

## Conclusion

This chapter summarises the thesis first, then proposes recommendations on how to control the discovered fraud and risk scenarios based on the study findings, and conclude with a future direction for this important area of research.

### 5.1 Summary of Thesis

This thesis has contributed to the existing knowledge on affiliate fraud, by examining and presenting multitude of ways to implement already known frauds and threats in AM. This enables researchers and practitioners to develop solutions that effectively mitigate all the discovered implementation variations of the known frauds. It also contributes to the existing knowledge having discovered new vulnerabilities that can be exploited by fraudsters in future. As researchers find ways to combat affiliate fraud, the fraudsters will continue to find newer ways to implement frauds that are already known or find new vulnerabilities that can be exploited. This research also fills another important gap, by introducing AMNSTE, a simulation and testing environment, as researchers do not often have unhindered access to “live” AM environments. The efficacy and utility of AMNSTE, to test fraud scenarios and to discover different methods of implementing a fraud have been demonstrated. Further, some practical solutions that can be implemented to prevent some of the risks have been described. Among others, the study has established how a reconcile application can be integrated between a third-party Affiliate Network and an advertiser’s back-office servers, to significantly reduce some of the known threats.

Examination of historical data of an AM practitioner helped in further analysing and uncovering possible fraud scenarios. This gave insights on the extent of fraud prevalence, from an advertiser’s perspective. The dataset does not allow us to quantify all affiliate fraud in a general sense, as major part of the dataset was filtered by the Affiliate Network using its fraud detection mechanisms. Nevertheless, the existence of 12% of fraudulent data among the filtered dataset,

that was assumed to be genuine by the AN, points us towards a high prevalence of fraud activity, in contrary to previous assumptions.

## 5.2 Recommendations for controlling mechanism against frauds and risks

It is an important decision for an AM practitioner to choose between an in-house development and an external third-party Affiliate Network, when evaluating the possibility of implementing an affiliate management system. In the case of an in-house tracking system, the system need to be developed using software components that are independent and loosely coupled with the transaction processing system, enabling the affiliate management system to update individual components regularly in response to the developments taking place in AM ecosystem. In contrary, third-party Affiliate Networks who do not have access to the transaction databases of the advertiser, can design their Web Services API with integration applications in mind.

Standard reconcile applications incorporating fraud-preventative features that are continuously updated in response to new fraud detections, can be made available to AM practitioners, that need to be hosted within AM practitioner's private network. They will be modular in design, hence can easily be customised to connect to any type of back-end transaction data store of an advertiser. While residing securely within AM practitioner's local network, the reconcile application can connect to the Affiliate Network via Web Services API.

Many of the solutions that can address the above discussed threats can also be implemented within the Affiliate Network. Some security checks, which do not slow down the tracking process, such as checking the *referrer* field, can be carried out real-time, each time a tracking information is processed, while other types of security checks, that are more resource-intensive, such as crawling affiliate sites to parse HTML, JavaScript and CSS pages for illegal code elements, can be implemented as independent scheduled processes, that get executed at regular intervals, such as on a daily or weekly basis. Load-time clicks can be controlled by requiring all the pages that are expected to carry a click-pixel to be registered with the Affiliate Network beforehand. Each time a click is tracked the *referrer* field of the click will be examined, which has to match one of the legitimate and registered

pages as the page of origin. A web-crawler can be used to crawl each of the pre-registered click-pixel bearing pages and any unregistered pages that might have caused clicks, at regular intervals. The crawler can expose any JavaScripts, iframes or other resource requests that are embedded in the crawled pages. If the crawler automatically receives any affiliate tracking cookies, as it crawls, it would indicate a load-time click fraud. No page shall serve a tracking HTTP cookie from the Affiliate Network, when the page is loaded, instead only when clicked on a tracking pixel. Sophisticated fraudsters have the capability to recognise such crawlers and react by not serving the usual cookie-stuffed page for crawlers, instead a legitimate page, to deceive the crawler's detection mechanisms. Chachra (2015) describes such situations and how the issues can be circumnavigated by adopting a number of methods, such as dynamically changing the referrer and user-agent fields of the HTTP request or using proxy servers to hide IP addresses, etc.

An integration application such as the one that we used for the above data analysis, that is hosted on the advertiser's server with the access to the transaction data stores can reconcile conversion data from the Affiliate Network, with the advertiser's back-end data store, which eliminates numerous kind of fraud attempts.

Using a reservation buffer model can further help to control conversion stealing by internal and external fraudsters. With the reservation buffer model two separate tables are maintained for reservation data. When a reservation is made on-line, at the time of the reservation, the reservation data is stored in the *Reservation buffer table*. The unique reservation identifier that is generated at this stage by the *Reservation buffer table* is provided to the customer and any other stakeholders that are external to the system, such as the third-party tracking process. At the time of processing the reservation, a new reservation identifier is generated and stored within the *Reservation table* that maps to the initial reservation identifier in the *reservation buffer table*. This new reservation identifier is used system wide internally, which remains hidden from external stakeholders to the system. This can be extended with payment buffer table that maps to payment table, etc.



During the case study, we came across some conversion entries within the tracking system, bearing reservation identifiers that are only used internally. This could represent an intrusion from an internal source who has access to the transaction database. Conversion entries for reservations with high values for which no commission has yet been paid, such as those resulting from organic searches or direct traffic can be credited in this manner to an existing affiliate account. Such observations alert us to internal intrusions. Though we did not find widespread evidence of internal intrusion, the biggest vulnerability that we came across, during our research is the hypothetical scenario of *Conversion Claiming*. During the process of developing the reconcile application, we recognised the capability of fraudulently claiming conversions that have not been claimed by any other affiliate. This can be done by any person with sufficient access rights to the advertiser's back-end transaction database. It can be implemented in most damaging manner by an internal software developer or a contractor who implements the reconcile software, by running an automated process as a scheduled task that selects "unclaimed" transactions such as direct sales, counter sales or sales that originated from organic searchers, etc. by running a query against the transaction database and then using the web services API of the Affiliate Network to assign those sales to a specific affiliate account of his choice. To avoid detection, the fraudulent developer may apply *business rules* either to select all unclaimed sales, or only some of the high-yielding sales, or a specific percentage of the unclaimed sales, etc. There is a real threat with regard to this fraud, as it is most likely, that many of the AM practitioners do not have the skills needed to implement a reconcile application, hence would hire an independent contracting software developer for the purpose. This fraud can be curbed by monitoring the tracking records that have been updated by using Web API. Similarly, an internal staff member who does not have access rights to update tracking database of the AN via Web API, can use an unauthorised web page within their own control, to "fire" the conversion pixel. Hence checking the *referrer* field and IP address as discussed above, can contribute to mitigating this risk.

It is evident that internal staff of a third-party Affiliate Network who has sufficient security privileges can add or edit bogus click- and conversion-tracking data, in

favour of an unscrupulous affiliate. This too is a high vulnerability for an advertiser, which can be easily managed by implementing the above mentioned solution of a reconcile application. As such members of an Affiliate Network do not have access to the back-end transaction databases of the advertiser, which is located within the advertiser's private network, access to authentic transaction identifiers and other transaction data located within the advertiser's databases, is restricted.

*Conversion faking* in the simplest form was observed in the dataset that we examined. A rogue affiliate can embed code for a *Load-time click* on a web page under the affiliate's control and cause the browser to receive a cookie and thereby create a click-tracking at the loading of the page. The conversion pixel code can be embedded on the same page with a bogus transaction identifier and total sum, to cause a fake conversion. The Affiliate Network platform will match the conversion record with the previous click record, but it is expected that the Affiliate Network should monitor the *referrer* field for each conversion record, which should be the URL of the confirmation page of the advertiser. It was observed that some records were a result of *Conversion faking* as the Affiliate Network has not implemented the above mentioned *HTTP referrer* check, and even a basic form of conversion faking with bogus transaction data would earn the affiliate a commission, provided that a reconcile application was not used by the advertiser. A more advanced form of the fraud would be, if the affiliate had access to a legitimate transaction identifier and a total price, in which case, the reconcile process will validate the transaction against the authentic transaction in the transaction database. Risk caused by the internal staff's access to transaction data, as discussed above, would be one source of access to legitimate transaction data, while a malware that runs as a browser plugin will be a source for an external threat.

The browser cookie can be used in different ways to achieve different outcomes, as per business rules. Under "Last person gets all" method, the last affiliate gets the full commission amount by overwriting any existing cookie placed by a different affiliate who promoted the same advertiser. Under "Share Commission by all" method, multiple affiliates can share the commission, either equally or proportionately, depending on the business rules. If "Share the commission by all"

rule is in place, the cookie on the visitor's browser does not get deleted; instead the last affiliate's identifier is added to the affiliates list within the current cookie. Another important business rule is implemented by deciding if the browser cookie is deleted at the end of each successful sales conversion. By deleting the cookie, the affiliate can earn a commission again from the same customer, promoting the same advertiser, as long as the customer's decision to buy an advertiser's product was influenced by an advertisement link placed in affiliate's website. Some advertisers consider that this method is appropriate, as if the advertiser has "won" the customer, after the first visit, then the customer would have visited the advertiser directly for subsequent purchases, thus eliminating the need of an affiliate recommendation. But the fact that the visitor was again motivated to visit the advertiser by the said affiliate, justifies the efforts of the affiliate to earn the new commission. With this business decision in place, any duplicate cookie at the time of conversion is a fraud scenario that can be easily detected. Such features are offered to the advertisers by the affiliate marketing networks, to implement advertiser's own choices as individual business rules. Those capabilities are unique selling points (USP's) of different Affiliate Networks.

Finally, the lifespan of the cookie is another important business decision that can be used as an USP. If the cookie never expires until the visit converts to a transaction, then the affiliate is assured of a commission for his effort, even if the visitor only returns to the e-commerce site, many weeks or months later to make a purchase. It is a way to honour the affiliate for his promotion of the sale.

### 5.3 Future Directions

Though there are contradicting results about the extent and prevalence of affiliate fraud (Chachra et al., 2015; Snyder & Kanich, 2015), the cases of Shawn Hogan and similar cases since then (Edelman, 2015) show that even a few unscrupulous affiliates can cause large-scale losses to advertisers. As researchers develop and strengthen the defensive capabilities of our systems, the fraudster and attackers continue to further refine their tactics to evade detection. Hence, continuous research efforts to stay ahead of the attackers will ensure the survival of this very effective on-line marketing model and its sustainability over time.

Most of the previous research have used different methodologies to quantify affiliate fraud, without access to real affiliate tracking data from Affiliate Networks or from Advertisers. This research has been able to analyse the affiliate tracking dataset of an advertiser, which gives valuable insight to prevalence of affiliate fraud in current ecosystem. Hence, further research in collaboration with more AM practitioners and third-party Affiliate Networks can shed more light on how widespread are affiliate fraud in different market segments and affiliate management systems.

Integration applications for reconciling tracking records with back-end transaction records of an advertiser, are an easy to implement, effective tool as the first line of defense, which is worth exploring. They can also function as effective Intrusion Detection Systems (IDS) and Intrusion Prevention Systems (IPS).

While serving as an effective mechanism to fight affiliate fraud, as exposed by this study, an integration application for reconciling records can be the cause of an even larger fraud, if the developers or integrators choose to embed malicious code, as discussed above, which should be further researched to curtail the risks involved with it.

# References

- Benediktova, B., & Nevosad, L. (2008). *Affiliate Marketing - Perspective of content providers*. Department of Business Administration and Social Sciences, Lulea University of Technology.
- Chachra, N. (2015). *Understanding URL Abuse for Profit*. San Diego, CA: University of California. Retrieved from <http://escholarship.org/uc/item/7nw0f6bf>
- Chachra, N., McCoy, D., Savage, S., & Voelker, J. M. (2014). Empirically Characterizing Domain Abuse and the Revenue Impact of Blacklisting. *In Proceedings of the Workshop on the Economics of Information Security (WEIS)*. Retrieved from <https://cseweb.ucsd.edu/~voelker/pubs/namevalue-weis14.pdf>
- Chachra, N., Savage, S., & Voelker, G. (2015). Affiliate Crookies: Characterizing Affiliate Marketing Abuse. *IMC '15 Proceedings of the 2015 ACM Conference on Internet Measurement Conference* (pp. 41-47). New York, NY: ACM. doi:10.1145/2815675.2815720
- Collins, S. (2011a). Finding Affiliate Programs. Retrieved July 27, 2015, from <http://www.extramoneyanswer.com/finding-affiliate-programs>
- Collins, S. (2011b). *Affiliate summit 2011*. Retrieved from ISSUU: <http://issuu.com/affiliatesummit/docs/affstat2011report>
- Dennis, L., & Duffy. (2005). Affiliate marketing and its impact on e-commerce. *Journal of Consumer Marketing*, 22(3), 161-163.
- Edelman, B. (2015). Retrieved from Affiliate fraud litigation index: <http://www.benedelman.org/affiliate-litigation>
- Edelman, B., & Brandi, W. (2015, February). Risk, Information, and Incentives in Online Affiliate Marketing. *Journal of Marketing Research*, LII, 1-12.
- Fiore, F., & Collins, S. (2001). *Successful Affiliate Marketing for Merchants*. Indianapolis, IN: Que Corp.
- Fox, P. B., & Wareham, J. D. (2012). Governance Mechanisms in Internet-Based Affiliate Marketing Programs in Spain. In I. Lee, *Transformations in E-Business Technologies and Commerce: Emerging Impacts* (pp. 222-239). doi:10.4018/978-1-61350-462-8
- Gregori, N., Daniele, R., & Altinay, L. (2013, June 18). Affiliate Marketing in Tourism: Determinants of Consumer Trust. *Journal of Travel Research*. doi:10.1177/0047287513491333
- Hoffman, D. L., & Novak, T. P. (2000). How to acquire customers on the web. *Harvard business review*, 179-188.
- Kanich, C., Kreibich, C., Levchenko, K., Enright, B., Voelker, G. M., Paxson, V., & Savage, S. (2008). Spamalytics: An Empirical Analysis of Spam Marketing Conversion. Virginia: ACM.

- Libai, B., Biyalogorsky, E., & Gerstner, E. (2003, May). Setting Referral Fees in Affiliate Marketing. *Journal of Service Research*, 5(4), 303-315.  
doi:10.1177/1094670503005004003
- Mariussen, A., Daniele, R., & Bowie, D. (2010). Unintended consequences in the evolution of affiliate marketing networks: a complexity approach. *The Services Industries Journal*, 1707-1722. doi:10.1080/02642060903580714
- McKnight, H. D., Choudhury, V., & Kacmar, C. (2002, Sep.). Developing and Validating Trust Measures for e-Commerce: An Integrative Typology. *Information Systems Research*, 13(3), 334-359.
- Meyer, M. H., & Webb, P. H. (2005). Modular, layered architecture: the necessary foundation for effective mass customisation in Software. *Int. J. Mass Customisation*, 1(1), 14-36.
- Performance Marketing Glossary*. (2015, May 28). Retrieved July 20, 2015, from Cake Performance Marketing: <http://getcake.com/performance-marketing-glossary/>
- Ray, A. (2001, August 23). Affiliate Schemes prove their worth. *Marketing - London*, pp. 29-30.
- Snyder, P., & Kanich, C. (2015). No Please, After You: Detecting Fraud in Affiliate Marketing Networks. *Workshop on the Economics of Information Security (WEIS)*. University of Illinois.
- Vacha, D., Saikat, G., & Yin, Z. (2013). ViceROI: Catching Click-Spam in Search Ad Networks. *CCS' 13*. New York, NY: ACM. doi:10.1145/2508859.2516688.
- Venugopal, K., Das, S., & Nagaraju, M. (2013, June). Business Made Easy By Affiliate Marketing. *Journal of Business Management & Social Sciences Research*, 2(6), 50-56.