

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.

**Improving the robustness and privacy of
HTTP cookie-based tracking systems
within an Affiliate Marketing context**

A thesis presented in fulfilment of the requirements

for the degree of

Doctor of Philosophy

at Massey University, Albany,

New Zealand.

Bede Ravindra Amarasekara

2021

Table of Contents

Glossary of frequently used terms and abbreviations	v
List of Tables.....	vi
List of Figures	vii
Acknowledgements	viii
Abstract.....	ix
Publications.....	x
Chapter 1 . Introduction.....	2
1.1 Research problem.....	7
1.2 Research Goals	8
1.3 Research significance	9
1.4 Research structure.....	11
1.4.1 Awareness of the problem	12
1.4.2 Suggestion process.....	14
1.4.3 Development and Evaluation processes	16
1.4.4 Conclusion process.....	16
Chapter 2 . Literature review	18
2.1 HTTP request and response	21
2.2 State management	25
2.3 Tracking on Internet.....	32
2.3.1 Affiliate marketing model.....	34
Stakeholders in AM.....	37
AM traffic generation models	39
Tracking process in AM	41
2.3.2 Business Analytics	48
2.3.3 Insights as a service.....	51
2.4 Stateless vs. Stateful tracking.....	51
2.4.1 Stateful tracking.....	52
2.4.2 Stateless tracking	52
2.5 HTTP cookies for tracking.....	54
2.5.1 Single-event tracking.....	55
2.5.2 Multi-event tracking.....	56
2.5.3 Single-domain tracking.....	57
2.5.4 Cross-domain tracking.....	57
2.6 Alternative tracking techniques.....	58

2.6.1 Flash cookies.....	59
2.6.2 Microsoft Silverlight	60
2.6.3 HTML5 Local Storage	61
2.6.4 ETag.....	61
2.7 Privacy concerns	62
Chapter 3 . Methodology	68
3.1 Selecting a research paradigm	69
3.2 Choosing the test environment.....	71
3.3 Hardware configurations	73
3.3.1 Hardware-based test environment.....	74
3.3.2 Virtual networking infrastructure-based test environment.....	75
3.3.3 Internet-based public test environment (Public-AMNSTE)	76
3.4 AMNSTE2 (System design and Implementation).....	77
3.4.1 Internet user / Researcher Domain	78
3.4.2 Affiliate website	79
3.4.3 Tracking service provider	84
3.4.4 E-commerce sites.....	86
3.5 Simulating privacy intrusions	87
3.5.1 Test case scenarios.....	88
AM model.....	88
Local Business insights gathering.....	89
Third-party Business Analytics offering.....	89
Chapter 4 . Artefact Description	91
4.1 Artefacts relating to alternative tracking methods	92
4.1.1 HTTP cookie-based tracking	94
4.1.2 HTML5 Local storage-based tracking	98
4.1.3 ETag-based tracking	102
4.1.4 Robust tracking	105
4.2 Artefacts relating to privacy models.....	110
4.2.1 Single domain tracking	111
4.2.2 Multi-domain tracking.....	111
4.2.3 Business insights gathering.....	112
Chapter 5 . Evaluation.....	115
5.1 First cycle.....	115
5.1.1 Using recursion to find UID candidates.....	116
5.2 Tracking failures.....	120

5.2.1 Fail scenarios.....	120
5.3 Defining a baseline for Tracking techniques	121
5.4 Evaluation of tracking capabilities tests.....	123
5.4.1 Evaluating test results	124
5.5 Privacy intrusion simulations.....	132
5.5.1 Tracking as an underlying technology.....	133
5.5.2 Tracking for information gathering.....	134
Business insights gathering experiment	135
5.5.3 Tracking by third-party business analytics services	136
5.6 CDN exposure	140
5.7 Tracking vector summary - utility, efficacy, and ease of use	141
5.7.1 Local Storage.....	142
5.7.2 ETag.....	145
Chapter 6 . Discussion.....	148
6.1 Research goal 1.....	148
6.2 Research goal 2.....	149
6.2.1 Internal tracking solution	149
6.2.2 External tracking solution.....	150
6.2.3 Single-event tracking.....	151
6.2.4 Multi-event tracking.....	152
6.2.5 Tracking vectors	152
Using Local Storage as a tracking vector	155
Robustness of ETag as a tracking vector	157
Versatility of Robust tracking	158
Respawning	159
6.3 Research goal 3.....	161
6.3.1 Information seeking behaviour.....	162
6.3.2 Tracking privacy model.....	172
6.3.3 Information scavenging.....	177
6.3.4 Tracking data spillage.....	178
6.4 Privacy and perceptions	180
Chapter 7 . Conclusions & future direction	183
7.1 Future direction	186
Appendix	187

Glossary of frequently used terms and abbreviations

AM	-	Affiliate marketing
AMN	-	Affiliate marketing network
AMP	-	Affiliate management platform
CDN	-	Content delivery network
CMS	-	Content management system
CORS	-	Cross origin resource sharing
CPA	-	Cost per acquisition
CPC	-	Cost per click
CPM	-	Cost per Mille
CSS	-	Cascading style sheet
CSP	-	Content security policy
DSR	-	Design science research
ETag	-	Entity tag
GDPR	-	General data protection regulation
GTM	-	Google Tag Manager
IETF	-	Internet Engineering Task Force
ISB	-	Information seeking behaviour
LSO	-	Local shared objects
NAT	-	Network address translation
OS	-	Operating systems
OSN	-	Online social networks
PII	-	Person identifying Information
SEO	-	Search engine optimisation
SME	-	Small-to-medium enterprises
UID	-	Unique identifier
URL	-	Uniform resource locator
XSS	-	Cross-site scripting
XDT	-	Cross domain tracking

List of Tables

Table 1: Publication schema for a DS research study (adapted from Gregor & Heiner, 2013)	17
Table 2: Common User-Agent identifier strings.....	23
Table 3: HTTP cookie attributes	27
Table 4: Stakeholders in Affiliate Marketing.....	38
Table 5: Visitor-traffic Generation Schemes, Pricing Structure, and ISB in an AM ecosystem	39
Table 6: Domain name and categories of Public-AMNSTE hosted on Internet	77
Table 7: List of HTTP cookie-based tracking processes	96
Table 8: List of Local storage-based tracking processes	99
Table 9: List of ETag-based tracking processes	103
Table 10: First partial list of tracking processes in a robust tracking scenario	107
Table 11: Second partial list of tracking processes in a robust tracking scenario.....	109
Table 12: PHP and .NET server variables	118
Table 13: Currency of tracking technologies	121
Table 14: List of experiments to check efficacy of tracking vectors.....	123
Table 15: Results of efficacy of alternative technologies as tracking methods	125
Table 16: IP address information	136
Table 17: Digital personas at each ISB level.....	174

List of Figures

Figure 1: Research process model (Kuechler & Vaishnavi, 2008)	12
Figure 2: Background resource requests made by browser	19
Figure 3: HTTP Request with headers.....	22
Figure 4: HTTP response with headers	25
Figure 5:Tracking process of the Affiliate Marketing Model	42
Figure 6: AMNSTE2 hardware-based network topology	75
Figure 7: AMNSTE2 network topology with virtual infrastructure.....	76
Figure 8: Affiliate home page.....	80
Figure 9: Links and descriptions of experiments.....	81
Figure 10: Tracking Technologies Group	82
Figure 11: Tracking results	83
Figure 12: Tracking Server Homepage.....	85
Figure 13: Sequence diagram for HTTP cookie-based tracking process.....	95
Figure 14: Sequence diagram for Local Storage based tracking process	98
Figure 15: Sequence diagram for ETag based tracking process.....	102
Figure 16: Sequence diagram for page-loading event using robust tracking process	106
Figure 17: Sequence diagram for click- and conversion tracking using robust tracking process	108
Figure 18: Multi-domain visitor tracking results	113
Figure 19: Click results in descending order	127
Figure 20: Last 10 conversion results in descending order.....	128
Figure 21: Results of Click-Tracking in descending order	133
Figure 22: E-commerce site gathering business insights locally	135
Figure 23: Multi-domain visitor tracking results	138
Figure 24: UID merged with social media account data	140
Figure 25: How "Off-Facebook activity" is sourced (Facebook, 2021b)	166
Figure 26: Google Timeline - Location history	169
Figure 27: Google Maps Timeline.....	171
Figure 28: Tracking privacy model relating to ISB.....	172

Acknowledgements

I express my heartfelt gratitude to Assoc. Prof. Anuradha Mathrani, who guided me throughout my post-graduate studies. Her inspiration and guidance led me not only to complete my PhD, but also how to be a scholar. I would not have been able to communicate my research through multiple conference and journal publications, without her guidance.

I am ever grateful to my co-supervisor Assoc. Prof. Chris Scogings, who paved the path for me to study at Massey University. Without your foresight, I could not have started this journey.

Finally, I wish to thank Massey University of New Zealand for awarding me a full doctoral scholarship for my entire duration of study; to Prof. Dianne Brunton, head of School of Natural & Computational Sciences for the conference and other grants, to all academics and administrators who made my academic journey enjoyable.

I dedicate this publication to my parents, family and to my Amarasekara and Subasinghe ancestors, for, without them I would not be here, today.

Abstract

E-commerce activities provide a global reach for enterprises large and small. Third parties generate visitor traffic for a fee; through affiliate marketing, search engine marketing, keyword bidding and through organic search, amongst others. Therefore, improving the robustness of the underlying tracking and state management techniques is a vital requirement for the growth and stability of e-commerce. In an inherently stateless ecosystem such as the Internet, HTTP cookies have been the de-facto tracking vector for decades. In a previous study, the thesis author exposed circumstances under which cookie-based tracking system can fail, some due to technical glitches, others due to manipulations made for monetary gain by some fraudulent actors.

Following a design science research paradigm, this research explores alternative tracking vectors discussed in previous research studies within a cross-domain tracking environment. It evaluates their efficacy within current context and demonstrates how to use them to improve the robustness of existing tracking techniques. Research outputs include methods, instantiations and a *privacy model* artefact based on information seeking behaviour of different categories of tracking software, and their resulting privacy intrusion levels. This privacy model provides clarity and is useful for practitioners and regulators to create regulatory frameworks that do not hinder technological advancement, rather they curtail privacy-intrusive tracking practices on the Internet. The method artefacts are instantiated as functional prototypes, available publicly on Internet, to demonstrate the efficacy and utility of the methods through live tests.

The research contributes to the theoretical knowledge base through generalisation of empirical findings and to the industry by problem solving design artefacts.

Publications

1) Security and privacy management in cross-domain tracking systems within an e-marketing context

Amarasekara, B.R., Mathrani, A., & Scogings, C. (2019). Presented at 6th IEEE Asia-Pacific Conference on Computer Science and Data Engineering (CSDE2019), Melbourne, Australia. DOI: 10.1109/CSDE48274.2019.9162393

2) Crookies: Tampering with Cookies to Defraud E-Marketing

Amarasekara, B.R., Mathrani, A., & Scogings, C. (2020). Published in Encyclopaedia of Criminal Activities and the Deep Web (3 Volumes). IGI Global Publishing. DOI: 10.4018/978-1-5225-9715-5

3) Improving the Robustness of the Cross-Domain Tracking Process

Amarasekara, B.R., Mathrani, A., & Scogings, C. (2020). Presented at 12th Asian Conference on Intelligent Information and Database Systems (ACIIDS2020), Phuket, Thailand. Proceedings published in Intelligent Information and Database Systems, Springer, Singapore. DOI: 10.1007/978-981-15-3380-8_23

4) Stuffing, Sniffing, Squatting, and Stalking: Sham Activities in Affiliate Marketing

Amarasekara, B.R., Mathrani, A., & Scogings, C. (2020). Published in Library Trends 68(4):659-678. Johns Hopkins University Press. DOI: 10.1353/lib.2020.0016

5) Online Tracking: When Does it Become Stalking?

Amarasekara, B.R., Mathrani, A., & Scogings, C. (2021). Published in Vietnam Journal of Computer Science. DOI: 10.1142/S2196888821500226

Chapter 1. Introduction

We live in a “connected” world, where a variety of devices (e.g., desktops, laptops, tablets, mobile phones, fitness trackers, etc.) form a part of our everyday lives. We move between locations (home, work, travelling) and use different devices seamlessly (desktop at work, laptop at home, mobile phone, and tablet). We start tasks in one location, continue with them at another, on a different device; as long as we are logged in to the services with the same account across all the devices, our interactions are synchronised across all devices, transitioning from one device to another, easily and smoothly. A product perused online, not purchased will appear during the days that follow, without having to search for it again. Many suggestions that appear as advertisements on webpages that we visit are often related to our current interests. All our online interactions are tracked and synchronised across devices, which might appear a great convenience to some users, but a dilemma to others who are worried about their privacy.

Tracking user-activities on the Internet is carried out by different parties for different reasons (Martin et al., 2003; Sanchez-Rola & Santos, 2018). The Internet is an inherently stateless ecosystem by design; HTTP cookie have been widely used to manage state since introduction in 1997 (Kristol & Montulli, 1997). Usually, during the first visit to a specific website, the webserver may store a unique identifier (UID) on the user’s computer, and additional data pertaining to the visit on a server-side datastore. During subsequent visits, the webserver can utilise this UID to retrieve a richer set of information specific to the current user, from a server-side datastore (Dwyer, 2009). Different entities track user activity for various purposes. E-commerce practitioners need a reliable tracking system to quantify and reward visitor traffic generators. Business analytic providers track user interactions to generate customer behavioural insight that assist targeted marketing capabilities (Castelluccia, 2012; Roosendaal, 2012). Governments and security agencies track user-activity to prevent national security threats. In this research, tracking within an e-commerce context is examined to narrow the scope of

research. The techniques that are described can also be applied to other contexts as an underlying technology.

The type and amount of information gathered during the tracking process differ, depending on the intended purpose and usage of tracking. Websites may track users and some of their personal information to enhance the user experience. During the interaction with a website, users might make specific choices such as the language, the currency of payment, a time zone, etc. and such user interface customisations can be saved by the website, avoiding repetition of these actions with each visit to the website. Such features are the default for websites today and are not seen as a privacy intrusion by users; rather an essential convenience, as it is intended for user's advantage. A website can go a step further and gather additional information such as what products were perused, or what was purchased among other data related to behavioural and preferential trends. With the help of this additional data websites can further customise user-experience, for subsequent visits. For instance, the home page can be composed of a product list with items that were perused, but not bought, hoping it would improve user-experience. If a subsequent visit was made to purchase a product that was previously perused, then the user would find the customised product list beneficial.

Though this kind of information gathering goes further than just saving user preferences, a user may usually not find it to be privacy-intrusive, but rather a convenience. The e-commerce site also benefits, by gaining the ability to use valuable customer information for target advertising, and business analytics that can lead to increased revenue generation. Additional person identifying information (PII) such as name, email and physical addresses can be added to the mix, by getting the users to fill out those details as mailing or delivery addresses. But using any of the PII for unsolicited advertising is much frowned upon by most users (Dwyer, 2009; Hoofnagle et al., 2012); though generally, the information gathered in a single-site tracking operation would not be considered privacy invasive, by many. Technological implementation of such a tracking system, within one single website

corresponding to one web domain is rather easy and straight forward. It involves placing a HTTP cookie in visitor's computer and reading it again on subsequent visits.

In contrast, tracking user-activity across multiple domains usually involves third parties and technological implementations that are more complex; therefore, most websites subscribe to specialised third party tracking services. Cross-domain tracking is useful for e-commerce, specifically e-marketing strategies, business analytics services that generate customer demographic data for business managers, security agencies that monitor criminal activities and national security threats across the web and researchers, alike. Business analytic services follow users across many websites or web domains owner by different establishments and gather data such as the origin of the web traffic by looking at the *referrer* field of the header, how long a user spend at specific sites or pages, what products were perused, what was purchased etc. Products that were peruses, but not purchased, gives a marketing lead to the re-marketers; amount of time spent in perusal gives an indication of product interests, which are valuable insights for marketers. The amount and comprehensiveness of the information to create a complete *persona*, depends on the level of visibility over the Internet, which are discussed in detail, in the discussion chapter. Search engines like Google or online social networks (OSN) like Facebook can combine above mentioned information with OSN accounts of the user to enrich with personal information (e.g., social, religious, political, and other affiliation, individual tendencies, motivations based on opinions extracted from social media posts), which can create a comprehensive persona, which is dangerously intrusive of personal lives of users. Such tracking has become synonymous with stalking in recent years (Hoofnagle et al., 2012). They may neither be for user's benefit, nor would have even had user's explicit knowledge or agreement (Baumann et al., 2019). While some of the information is shared with the website owners, such as business analytics, in most cases, the type of information and the level of privacy intrusion is even opaque to the website, with whom the user interacts. Such third-parties gather browsing history and user activity data, which are then combined with personal data of the user to create customer

demographics that can be used for target marketing for commercial gain (Libert, 2015). Information that offer business insights have become very popular among business managers, who subscribe to Business Analytics services such as Google Analytics. WikiLeaks and recent Cambridge Analytica scandal have exposed the prevalence of this kind of tracking activity (Berghel, 2018; Laterza, 2018; Manokha, 2018; Margaret, 2020; Richterich, 2018; ur Rehman, 2019). Such services can range from AM systems, business analytic providers such as Google Analytics, utility or widget providers for visitor counters, weather or currency information services, and content delivery networks (CDN). While the user information thus gathered do not directly improve the user-experience, and therefore not directly beneficial for a visitor, it is often also opaque to visitors of the site. Nevertheless, in an ever-evolving Internet ecosystem, cross-domain tracking has become a technical necessity underlying e-commerce and e-marketing efforts.

Cross-domain tracking capabilities are vital for e-commerce activities. Visitor traffic to e-commerce sites is usually generated outside of the e-commerce domain. Search engine advertising, improving organic search traffic through search engine optimisations (SEO) and AM are key traffic generators. Under AM model, e-commerce sites sign up independent websites called “affiliates”, who already have a wide reach of the type of site visitors who can become potential customers for the e-commerce site. Affiliates earn a monetary reward from advertisers for the visitor traffic they generate towards the advertisers’ e-commerce sites. AM is considered the most cost effective advertising, as well as traffic and revenue generation model on the Internet (Brear & Barnes, 2008; Norouzi, 2017). While AM is vital for anyone engaged in e-commerce, it has become a lifeline for small-to-medium enterprises (SME) who have a web presence. SMEs usually subscribe to an Affiliate Network, which is a third-party tracking technology provider; this research uses the term Affiliate Management Platform (AMP) for clarity. Larger AM practitioners such as e-bay and amazon.com manage their tracking processes, in-house.

Different advertising models, sometimes called compensation models, are used in AM. Cost-Per-Click (CPC) and Cost-per-Mille (CPM) methods were initially popular, where an affiliate is paid for each “click” (visitor) under CPC, or for each display of a banner advertisement under CPM. But both advertising models are losing popularity as they are plagued by click-fraud. Cost-Per-Acquisition (CPA) model appeared as the silver-bullet against click fraud(Hu et al., 2013). CPA solves two main problems that marketers encounter on a regular basis, in online marketing or other traditional marketing scenarios. These are:

- I) Advertisers have to spend their marketing budget upfront, to attract potential customers
- II) The potential customers thus accrued, might not result into desired outcomes such as purchasing goods or services, signing up for memberships, etc., thereby wasting marketing budget on an unintended target market.

With CPA advertising model, the advertisers pay affiliates a commission or a fixed amount, only for business outcomes, for example: if a customer purchases goods or services, but not for site visitor traffic or “clicks”. Under CPA marketing model advertising costs guarantee a desired outcome. Secondly, the advertising cost is paid “after” the purchase, not beforehand, unlike in other marketing models(Norouzi, 2017).

The discovery of “cookie-stuffing” fraud has indicated that CPA is not immune to fraud either, though far less than the frauds faced by CPC and CPM models (Amarasekara & Mathrani, 2015; Chachra, Savage, & Voelker, 2015; Edelman & Brandi, 2015; Snyder & Kanich, 2015; Vacha, Saikat, & Yin, 2013). The litigation against Shawn Hogan by e-bay in 2013 for AM fraud of over 15 million US dollars indicate the volume of some AM fraud (Edelman, 2015).

1.1 Research problem

Though HTTP cookies have been providing a reasonably reliable tracking capability, previous research have shown instances when the tracking systems can fail. Some of the failures are technical glitches while others are results of fraudulent manipulations undertaken by rogue actors to skew the tracking results (Amarasekara & Mathrani, 2017). Tracking failures are discussed in section 5.2.

Some alternative tracking vectors that provide better outcomes have been discussed in previous research. *Super cookies* dominated discussions as an indestructible tracking vector which can even re-spawn cookies that have been deleted (Soltani et al., 2010). Others have discussed the possibility of using HTML5 local storage and ETags, which is a short name for *Entity Tags* (Ayenson et al., 2011). As none of these alternative vectors have been originally designed for tracking purposes, they can cease to function after some time. The literature review shows new research studies (Yang & Yue, 2020), which still discuss flash cookies and super cookies and the resulting privacy threats, while we know that they are deprecated technologies. E-commerce and e-marketing are important research topics outside of Information Science (IS) research domain too (e.g., for business and social science research domains). They do not have the capacity to verify through experiments, hence depend on IS research outputs. Therefore, there is a need to assess the efficacy and utility of alternative tracking vectors as of now, and to update the current knowledge base.

Some previous research findings also demonstrate the use of alternative tracking vectors, within single-event and single-domain tracking scenarios as proof of concept. E-commerce activities usually need the capability to track user-activity across multiple events and multiple domains, which is much more complex. The “single-origin” concept in web security implementations render many information accessing and sharing capabilities that function well within a single domain, inaccessible across different domains.

Those research findings discuss the prevalence of multiple alternative tracking vectors, through empirical observations of their use, but implementation details under different tracking scenarios which are valuable to practitioners are lacking. As such implementation details sometimes deviate from or even contravenes the IETF technical specifications, it is a gap in the useful knowledge for developers, which this thesis addresses in chapters 5 and 6.

The literature review section provides technical, social, and psychological perspectives of online tracking and privacy. Tracking practices can range from unobtrusive, non-PII based tracking scenarios to those that gather complete digital personas. Governments and regulatory bodies are increasingly implementing new laws to protect privacy, but many of them fail to address the issue well (Matte et al., 2020; Papadogiannakis et al., 2021; Sanchez-Rola et al., 2019; Utz et al., 2019). Due to lack of clarity and knowledge as to what practises constitute to privacy invasiveness and the underlying tracking techniques, they can inadvertently curb the advancement of technology. A privacy model that demonstrates the correlation between its constructs, i.e., the type of application (and its information seeking behaviour), the level of privacy intrusion, the technical implications (such as single-domain vs. cross-domain tracking), will benefit research studies, software developers and practitioners as well as regulators to identify, classify and target tracking practices specifically.

1.2 Research Goals

My research goals are threefold:

Goal 1: Evaluate the currency of the tracking vectors through live experiments

Goal 2: Design a solution to improve the robustness of HTTP cookie based and privacy-preserving tracking process

Goal 3: Develop and verify through experimentations a privacy model based on levels of privacy intrusions.

1.3 Research significance

Alternative tracking technologies have been presented and discussed in research literature over the past decade. *Flash cookies* were discussed by since 2009 (Soltani et al., 2010); Entity Tags (ETags) and Cookie respawning techniques were demonstrated by Ayenson et al. (2011); Mittal (2010) evaluated DOM storage together with other tracking vectors. Many research works have been carried out in the past decade discussing these tracking techniques; privacy and online security related research viewed them as privacy intrusions, while business and enterprise related research works discussed those same techniques as opportunities to support e-marketing and web traffic generation systems. As business research is not involved in testing technological utility or currency of those tracking vectors, new research works continue to appear with the assumption that those techniques are still current (Yang & Yue, 2020).

HTTP cookies that are part of the HTTP protocol, therefore are meant for state management purpose. They can be expected to remain current, though other alternative tracking vectors were not designed as tracking vectors, therefore may lose their tracking capabilities with future development of those technologies. A regular evaluation of their currency needs to be undertaken by researchers in software engineering and related fields. The literature review has not revealed any recent research work that evaluated the currency of the alternative tracking vectors, nor design and implementation aspects of such techniques within a *Design Science* paradigm. This research is expected to fill this knowledge gap, providing researchers and practitioners answers to the questions: Which of the alternative tracking vectors are still functional and which techniques have ceased to exist? Which techniques can be used for which tracking scenarios (e.g., single-event, multi-event, cross-domain)? What are the technical design and implementation details?

As tracking is fast becoming synonymous with *stalking*, and many countries are introducing new legislature to protect user privacy. This research provides proof-of-concept, how web traffic generation and e-commerce activities can be implemented in a privacy-preserving manner.

The experiments are designed around an AM context, which involves multi-domain tracking system, which utilises single-event and multi-event tracking techniques. This enabled experimentation of diverse tracking scenarios within one experimental setup. The knowledge that was gathered and the techniques that were identified can be applied to many other e-commerce and Internet-based scenarios. This study therefore investigates the underlying technology that tracks user activity reliably across multiple domains, wherein many e-commerce activities occur. As cross-domain tracking technology has a great impact on privacy concerns, on espionage concerns and on IT security in general, the results of this research will be helpful in future research efforts in the above areas of study, too.

Brear and Barnes (2008) find that some retailers generate as much as 65% of their sales through AM. Therefore, apart from the benefits to the scientific and research community, this research will help to maintain the viability of the most cost-effective marketing model available to SME's and also to every e-commerce practitioner in general by developing solutions to mitigate fraud and vulnerabilities. The research outcomes and recommendations arising out of this study will also allow the industry to make informed decisions in implementing business insight gathering technologies and tools, thus avoiding information security breaches. The exploratory nature of this research will contribute significantly to the existing knowledge base.

The following chapters are organised as follows: In the next chapter - *Literature review* – the currently available research knowledge on this topic is discussed. The reviews lead the reader to the *methodology* that was used for this research and the justification on the choice of design science paradigm. Next, the iterative process undertaken to find plausible methods to recognise a user uniquely on the Internet, identifying which techniques were useful and which were not are presented. The *Artefact Description* chapter presents a detailed description of each chosen tracking vector implementation along with a process diagram. In the *Evaluation* chapter, the test results that helped validate study findings are shown. In the discussion section, generalisation of research findings within

a wider context is discussed, which contributes to the theoretical knowledge base. The research conclusions with suggestions for the future direction of this research are then presented.

This research attempts to solve a real problem that exists in the industry, that has a great impact due to huge financial losses through fraudsters. It also aims at providing clarity on privacy issues in tracking methods. With the initial analysis of the research problem, it became clear that design science paradigm is best suited for this research, which is a paradigm meant for solving existing problems in the industry. This choice will be discussed in further details in the methodology section.

1.4 Research structure

Activities of a design science research are grouped into different phases. Figure 1 presents the five phases suggested by Kuechler and Vaishnavi (2008). It aligns with the six-step process model by Peffers et al. (2007), which consists of:

(1) identify problem, (2) define solution objectives, (3) design and development, (4) demonstration, (5) evaluation, (6) communication.

The model by Kuechler and Vaishnavi (2008) combines steps three and four of the above model, as one single phase named *development* (Figure 1). The initial design from step two, goes through a rigorous cycle of build and evaluation processes iteratively, until the built artefact is evaluated as fit for the purpose.

The flow of activities within this research are described below, based on the model of Kuechler and Vaishnavi (2008)

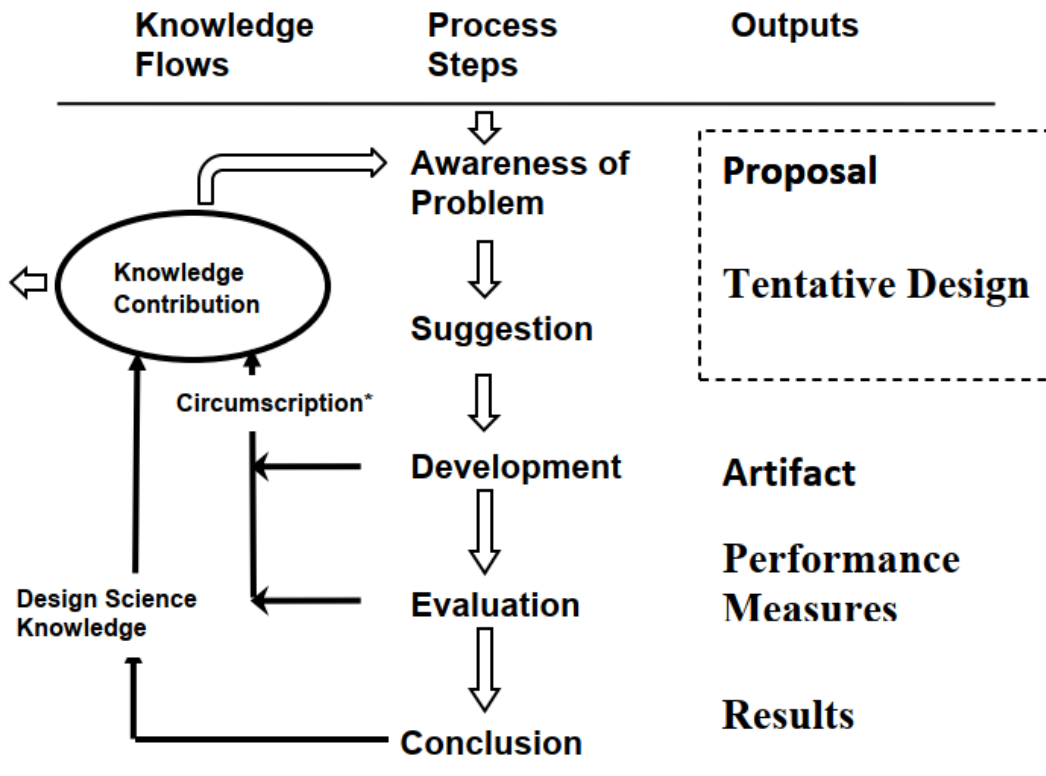


Figure 1: Research process model (Kuechler & Vaishnavi, 2008)

1.4.1 Awareness of the problem

During a Software Development contract undertaken in 2015 for a tourism service provider, the thesis author was responsible in implementing an AM program, that subscribed to an existing overseas Affiliate Marketing Platform (AMP). Reports based on web traffic generation indicated possible fraud scenarios, which led to the design and implementation of a transaction reconciliation application, which reconciled AMP generated web traffic and transaction data with the real back-end transaction database. The frauds discovered created the awareness of the need of more robust tracking capabilities.

As part of the above software development contract, the author also implemented Google Analytics service that provided business information needs to the marketing management team. During the process, it provided insights into the inadvertent information leakage to third parties, that occur through such implementations. The problem appeared even more acute, when *Google Tag Manager* (GTM) service was used to trigger the conversion-tracking process of the AMP, as it exposed all the

sales data to and additional third-party. Revealing all monetary conversions (customer purchases) to an AMP, which is also a third-party, has its own associated information risks. Exposing the data to another third-party such as Google, by using GTM to unify all trigger management activities in one place, was a convenience for less tech-savvy marketing teams, who did not understand the associated risks and data security breaches involved.

Further, empirical observations made it evident that some management decisions made by marketing and strategic management teams at small to medium enterprises (SME), who lack technical expertise, may routinely expose critical business information, inadvertently. Most SMEs usually do not have a Research and Development (R&D) department; therefore, the duties and obligations of a contracted developer did not allow the time and the resources that were required to investigate those problems that were identified. This awareness of the problems discovered in the industry led the author to consider this research topic for the PhD program. A comprehensive literature review was undertaken after formulating the initial research problem, based on prior industry experience. During literature review on AM, the financial impact of affiliate fraud leading to millions of dollars were discovered (Edelman, 2015; Edelman & Brandi, 2015). Further, while click-fraud has widely been discussed in available literature, sparse amount of literature focuses on conversion-fraud, thus indicating a knowledge-gap in the understanding of risk and fraud in performance-based marketing models, that depend on click- and conversion-tracking. The limited knowledge available has mainly focused on the outcomes and financial impact due to the affiliate fraud and further a few fraud methods have been named (e.g., cookie-stuffing) and described at a general level (Edelman & Brandi, 2015).

Chapters 1 and 2 capture activities of the *Awareness of Problem* process. As the output of this process a well formulated research problem with a clear set of research goals were developed. The research problem broken down to research goals lay the foundation to generate artefacts that fulfil solution requirement for the given problem-space. The artefacts thus created provide a theoretical contribution to the knowledge base.

1.4.2 Suggestion process

During this process constructs were identified and operationalised. The variables and value ranges that will be measured through the experiments determined. Two tracking construct groups were identified:

i. **Tracking construct group:**

The concept of tracking was represented by the Tracking construct group. Each of the three tracking constructs have a variable named *Result* with a dichotomous value of “true/false” for the capability measured. A “true” result confirms a tracking event, which is captured at the tracking server, with associated tracking data, and a verifiable UID is placed into or received from client system (e.g., a HTTP cookie with an UID).

- a. **Tracking:** This measures a successful tracking event with associated tracking data captured at the tracking server. If the tracking is captured, the associated data will also be accurately captured, as the data is part of the HTTP-request headers.
- b. **Click-tracking:** This is measured as successful, during web traffic generation experiments. A user-click or a successful display of a banner advertisement on captured on the tracking server returns a *success* result.
- c. **Conversion-tracking:** This is also measured during AM related experiments. A payment action at an e-commerce site that is being captured at tracking server, should be able to reconcile with an existing click-result, to return a *success* result.

ii. **Privacy construct group:**

The privacy concept was represented by the privacy construct group. The single construct within group, the user-privacy construct has multiple variables. are measured to ascertain

the level of intrusion a variable named *Result* with a dichotomous value of “true/false” for the capability measured.

- a. User-privacy: The “PII” variable has a simple dichotomous value. *True* value represents a tracking instance exposing a user in a privacy intruding manner. It can include contact details (Name, E-mail, phone, address) or any of OSN handles (Facebook, Twitter, Google profile URLs or handles). A “False” result indicates tracking a user with a non-person identifying ID.

The “ISB Level” variable’s value ranges from 1 to 5 based on the *information seeking behaviour* of the tracker, defined by the *Privacy Model* presented in *Discussion* chapter (Figure 28). The lower value represents less privacy intrusiveness in the tracking process, usually carried out as a technological necessity by an e-commerce operator. Higher values represent privacy-invasive tracking by an OSN, browser or operating system (OS) manufacturer.

A suitable methodology was established; design science paradigm was best suited to develop a solution following an iterative design-evaluation and necessary tools identified. As most of the experiments are security invasive by nature and will be subjected to severe restrictions if carried out on real Internet domains, two design options were evaluated for creating a routed network. Using CISCO packet tracer application, a fully functional virtual network was designed. A prototype was created using a virtual network, and basic functionality of the server software was developed for a functional prototype. The virtual design was then translated to a physical network comprising of physical hardware devices, including servers, routers, and switches.

Chapter 3 *Methodology* captures the *Suggestion* process. As outputs of this process constructs were operationalised, experiments and scope were defined, and a physical and virtual multi-domain network designed and implemented as a simulation environment for experiments.

1.4.3 Development and Evaluation processes

Through an iterative development and evaluation process, a functional prototype was developed, which demonstrates a tracking capability with improved robustness. During the process tracking vectors mentioned in previous research works were evaluated for their efficacy and utility within current context, which added to the existing knowledge base. Experiments with chosen tracking vectors were carried out iteratively, to refine the developed artifacts were fit for the purpose.

Chapters 4 *Artefact description* and chapter 5 *Evaluation* capture the *Development and Evaluation* process. As outputs of this process chapter 4 presents a concise description accompanied by a sequence diagram, of software artefacts capable of tracking using different tracking vectors, and for an improved tracking capability by combining different tracking vectors.

1.4.4 Conclusion process

In the previous chapter 5, the results were evaluated within the experimented application context. Chapter 6 Discussion further generalised those findings, as it applies to a wider context, making Theoretical contributions to the knowledge base.

The above five processes relate well with the publication schema suggested by Gregor and Hevner (2013) as presented in Table 1. The rest of the thesis follow this schema.

Table 1: Publication schema for a DS research study (adapted from Gregor & Heiner, 2013)

Section	Contents
1. Introduction	Research problem introduced and research significance defined. Key concepts on web and tracking. Overview of methods, constructs, research structure and the structure of the remainder of the paper. Three research goals drives the research, specifying requirements of the artefacts, and outcomes.
2. Literature Review	Background on web, tracking technologies and web traffic generation model. Prior research on tracking vectors that need evaluation. Findings from own previous research on AM frauds that define requirements to prevent them.
3. Method	Need for a simulation environment, design, and development thereof. Define experiments.
4. Artefact Description	Implementation details of each chosen tracking vector, on its own and in combination, to create a robust technique. Presented as sequence diagrams and process descriptions to provide sufficient abstraction and prescriptive knowledge to make new contributions to knowledge base.
5. Evaluation	Tracking results discusses to provide validity, utility, quality, efficacy, and fit-for-purpose.
6. Discussion	Results as outcomes based on three research goals, interpreted in a general context, making them useful for a wider application context. A privacy model based on experiment findings presented, that gives clarity to privacy intrusion levels based on information seeking behaviour of different tracking practices.
7. Conclusions	Emphasis on key findings, limitations, and future direction.

Chapter 2. Literature review

In the early days of the World Wide Web (WWW), webpages were primarily focused in delivering static content, usually as a document with some multimedia content such as images. The WWW and the associated Hypertext Transfer Protocol (HTTP) is a typical *stateless* client-server model, where the client browser initiates a connection and requests a resource from the webserver; after the webserver has served the requested resource, the request is deemed complete, and no *state* is maintained. Each request is considered a new request. But with the advent of dynamic Web 2.0 and increased diversification of the use of Internet with expanding e-commerce activities required a state management mechanism. HTTP cookies were proposed to provide this capability, which allowed websites to save user preferences locally on the user's computer, in a HTTP cookie (Kristol & Montulli, 1997). Netscape has been using cookies since 1994 on Mosaic browsers, but the Internet Engineering Task Force (IETF) standardised and published RFC 2109 under *HTTP state Management Mechanism* in February 1997 (Kristol, 2001).

A webpage usually consists of many different resources, some are visible to the user (e.g., text and images), while others are not directly visible components (e.g., headers, scripts, style sheets), which have a support function. For instance, all the textual components that make up the visible part of the webpage may be embedded in the HTML page, which is requested by the client browser from the webserver within the visiting domain. The HTML code within the page, is likely to have links to Cascading Style Sheet (CSS) files and JavaScript files that are not visible components, but the former defines how the visible components should be arranged on the page, while the latter executes code to add specific functionality to the page. In addition, there are image, video and audio files linked within the HTML code of the webpage. Such components may either reside locally within the same domain as the requested webpage, or they may link to external sources, within other web domains. When the client browser loads the webpage, it parses the HTML code, and issues further web requests

for each of the embedded resource. The user does not see these additional web requests that the browser makes in the background from local servers within the domain that the user is visiting, and from those external domains; the user has no control over this. Figure 2 shows the webpage returned by the server when the Uniform Resource Locator (URL) <https://www.google.com> is visited on a Microsoft Edge browser. The empty and simple Google home page only shows the Google logo, a search box, and a few hyperlinks. But the browser in *developer mode* reveals forty HTTP resource requests made in the background unbeknown to the user, resulting in browser receiving 1.9 MB of resources from various web domains. Some requests are from “google.com” domain that was

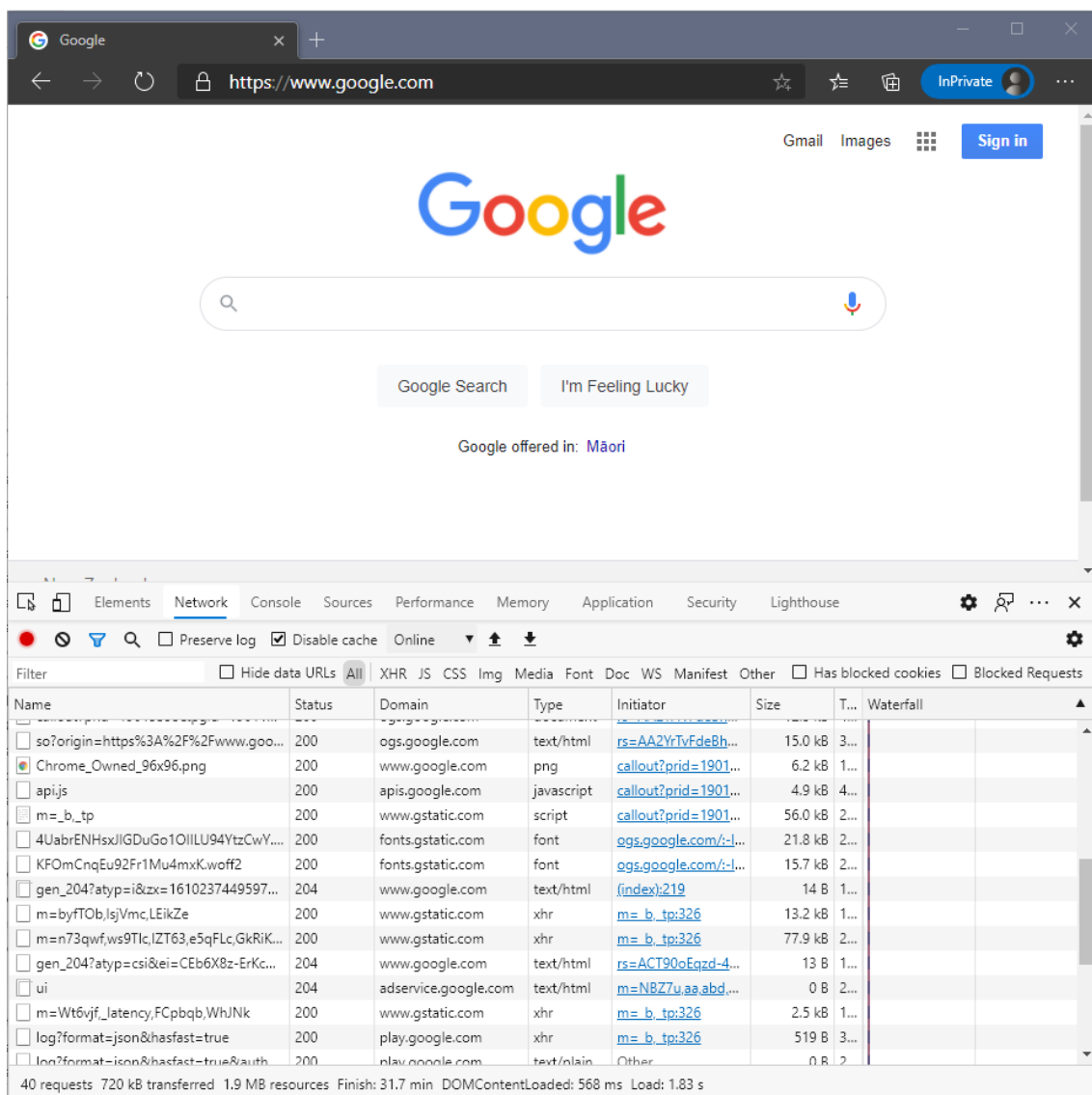


Figure 2: Background resource requests made by browser

visited in this example, some from sub-domains i.e., play.google.com, ogs.google.com, apis.google.com, adservice.google.com, while others are from external domains such as gstatic.com and googleusercontent.com, neither of which were intentionally visited by the user. This research will investigate this phenomenon and its influence on tracking capabilities as well as the security implications, through experiments.

While the domain that owned the requested webpage may send a cookie (first-party cookie) to keep track of the user, other domains that were not directly visited by the user, but whose content were requested and loaded in the background by the browser, might also decide to send a cookie with the requested resource (third-party cookie). In other words, if the cookie belongs to the same domain as the requested web page, such cookies are called “first-party” cookies. If the cookie belongs to a different domain than the web page, then it is a “third-party” cookie. Third-party cookies are usually placed by business analytics services or advertising service providers (Eckersley, 2010; Hoofnagle et al., 2012). Third-party web advertising companies and business analytic service providers often offer attractive services, widgets, and other web components that attract web authors to link their webpages to such third-party servers, so that a cookie or even a JavaScript file can give access to user’s browsing behaviour (Castelluccia, 2012; Dwyer, 2009; Libert, 2015). Browsers by default allow third-party cookies to be received, and even encouraged. When a user disables third-party cookies, the browsers usually warns that some features may not function correctly, which discourages users from disabling them.

But they can be a security risk (Kristol, 2001). Early IETF working groups discussed this phenomenon as *verifiable* and *unverifiable* transactions. A HTTP transaction is verifiable if the user can review the request URL before the transaction: i.e., when the user types the URL in to the address bar of the browser or places the cursor on a hyperlink and verifies the underlying URL in the status bar of the browser. But the resource URLs embedded in the HTML code of a webpage are *unverifiable transactions*, as the user does not have an option to not load them or access those webservers. Thus,

browsers use *unverifiable transactions* when sending cookies, loading resources embed in a HTML page, and makes redirections. Though users may expect to find cookies from websites that they have previously visited, most users are shocked to find cookies of websites that they never visited, stored within their computers (Kristol, 2001, p. 31).

2.1 HTTP request and response

A client application that requests a resource such as a webpage from a webserver identifies itself with a unique identifier string within the *User-Agent* header. As the most common client application in use is the web browser, in this thesis, the term “client browser” is used in place of *User-Agent*, for clarity. An HTTP resource request sent by a client-browser to a webserver, and the HTTP response from the webserver back to the client browser has three parts to each response. Figure 3 and Figure 4 show the HTTP request sent and the received, respectively.

1. The request (or response) itself makes up the first line in each image.
2. Rest of the lines make up the request (or response) headers
3. Not shown here but accompanied by these request (or response) headers are the content, that make up the body, displayed in the browser.

The GET verb in the first line makes a GET request from the server path “/alternate/TrackRobust” using HTTP protocol version 1.1 (Fielding & Reschke, 2014). Each line that follows represent a *request header* field and its value sent by the client browser with the HTTP request to the server. The RFC7231 by IETF defines headers and values it may contain (Fielding, 2014). While the headers are self-explanatory, some of them that are noteworthy and discussed next. They appear as good contenders for experiments in this research, in the quest to identify users uniquely, to improve the tracking process.

```
GET /alternate/TrackRobust/ HTTP/1.1
Host: connex.net.nz
Connection: keep-alive
Pragma: no-cache
Cache-Control: no-cache
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/87.0.4280.88 Safari/537.36 Edg/87.0.664.66
Accept: */*
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: cors
Sec-Fetch-Dest: empty
Referer: https://connex.net.nz/alternate/GetClickPixelRobust
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9
Cookie: connex=637396838417267094
```

Figure 3: HTTP Request with headers

User-Agent header

This header uniquely identifies the application that is making the HTTP request, which is in most cases a web browser. Table 2 shows some of the *User-Agent* strings that currently occur. Usually, a unique name for the browser followed by an optional version number separated by “/”, as in “Mozilla/5.0” above. *User-Agent* fields today have multiple product identifiers with version numbers in one string, denoting compatibility with different known issues. By convention, the product identifiers are listed in decreasing order of their significance for identifying the client software. This allows a webserver to identify the client browser and any known limitations or compatibility issues related to the client, which will enable the webserver to tailor its response taking those issues in to consideration. Client applications other than web browsers, such as web crawlers that are used by search engines to index websites (e.g., Googlebot, Bingbot, etc.) and web-scraping applications, have their own unique *User-Agent* identifiers; webserver can respond to those applications differently than to a web browser. As discussed later in the chapter, this header alone does not allow a user to be identified uniquely, since millions of browsers of a specific product and version, will use the same identifier; but in combination with other information contained in the client request, it can provide some uniqueness.

Table 2: Common User-Agent identifier strings

Browser/Crawler/Bot	User-Agent identifier string
Chrome	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/87.0.4280.88 Safari/537.36
Edge	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/87.0.4280.141 Safari/537.36 Edge/87.0.664.75
Internet Explorer	Mozilla/5.0 (Windows NT 10.0; WOW64; Trident/7.0; rv:11.0) like Gecko
Firefox	Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:79.0) Gecko/20100101 Firefox/79.0
Opera	Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/87.0.4280.88 Safari/537.36 OPR/73.0.3856.329
Googlebot	Mozilla/5.0 AppleWebKit/537.36 (KHTML, like Gecko; compatible; Googlebot/2.1; +http://www.google.com/bot.html) Chrome/87.0.4280.90 Safari/537.36
Googlebot (mobile)	Mozilla/5.0 (Linux; Android 6.0.1; Nexus 5X Build/MMB29P) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/87.0.4280.90 Mobile Safari/537.36 (compatible; Googlebot/2.1; +http://www.google.com/bot.html)
Bingbot	Mozilla/5.0 (compatible; bingbot/2.0; +http://www.bing.com/bingbot.htm)
Lightspeed crawler lightspeedsystems.com	LightspeedSystemsCrawler Mozilla/5.0 (Windows; U; MSIE 9.0; Windows NT 9.0; en-US)
Baidu	Mozilla/5.0 (compatible; Baiduspider-render/2.0; +http://www.baidu.com/search/spider.html)
Baidu (mobile)	Mozilla/5.0 (Linux; U; Android 8.1.0; zh-CN; EML-AL00 Build/HUAWEIEML-AL00) AppleWebKit/537.36 (KHTML, like Gecko) Version/4.0 Chrome/57.0.2987.108 baidu.sogo.uc.UCBrowser/11.9.4.974 UWS/2.13.1.48 Mobile Safari/537.36 AliApp(DingTalk/4.5.11) com.alibaba.a

Referer (sic) header

This header points to the URL that caused the current HTTP request to happen. If a user typed the URL into the address bar of the browser, or chose from a saved bookmark, the HTTP request does not have a *Referer* field. If the current request is caused by a click on a hyperlink, then the URL of the last page is indicated by the *Referer* header. The *Referer* can contain a URL within the current domain, if the navigation is within local domain; but it is also very useful to track users arriving from external domains. If the request is for a resource embedded within a webpage, such as shown in the headers of the Figure 3 above, *Referer* field refers to the HTML parent page, which has caused this resource request.

This header is important in e-commerce and e-marketing applications to ascertain the source of the user traffic. If the traffic is supposed to have been promoted by an affiliate, the *referrer* field should point at the specific affiliate's landing page. We can also experiment on serving different pages or reacting differently to client requests based on the origin of the traffic indicated by this field.

Cookie and Set-Cookie headers

The *Cookie* header is used in *HTTP requests* to return a cookie to the webserver. In contrast, the *Set-Cookie* header is used by the *HTTP response* to request a browser to save a cookie in its cookie store. The *Cookie* header on the last line on Figure 3 indicates that the browser has already got a cookie named *connex* in its cookie-store, received during a previous visit to the webserver, and the browser is returning the cookie. Figure 4 below, shows the HTTP response from the webserver which contains a *Set-Cookie* header that is used to instruct the browser to set a new cookie. In a traditional usage, the webserver will use the value of the *connex* cookie to customise the response specific to the user, and send it back to the browser, without the need to set a new HTTP cookie. As long as the cookie is not expired, the client browser will always return that cookie to the webserver in all future resource requests and the webserver can continue to use the identifier value to customise the response to the user. A server may send a new cookie with its *expires* attribute set to a past date, to delete a cookie

from the cookie storage of the browser. The server may also decide to change or add additional data to the cookie to be stored, such as keeping a count of number of visits or the last visit etc. But in this instance, the webserver is setting a new cookie each time with the *Set-Cookie* header as seen on Figure 4, to ensure the expiry date of the cookie extended by another three months. This type of strategy is adopted by web servers to track their regular visitors only, those who at least visit once in three months.

```
HTTP/1.1 200 OK
Cache-Control: private
Content-Type: text/plain; charset=utf-8
Content-Encoding: gzip
ETag: "ET637396838417267094"
Vary: Accept-Encoding
Server: Microsoft-IIS/8.5
X-AspNetMvc-Version: 5.2
X-AspNet-Version: 4.0.30319
Set-Cookie: connex=637396838417267094; expires=Fri, 02-Apr-2021 02:46:30 GMT; path=/; secure; HttpOnly; SameSite=None
X-Powered-By: ASP.NET
X-Powered-By-Plesk: PleskWin
Date: Sat, 02 Jan 2021 02:46:30 GMT
Content-Length: 139
```

Figure 4: HTTP response with headers

2.2 State management

During a browsing session, the state can be managed in different ways. Though HTTP cookies are a popular mechanism, embedding state data as parameters within the URL or within a hidden field of a form are other possible methods. But when utilising such methods, *state data* does not become part of the HTTP protocol; therefore, is prone to failure. The successful transfer of state between server and client rests upon the application developer. It involves explicit intervention of server application and client browser to send data and receive state data. In contrast, an HTTP cookie is part of the underlying protocol. The server application simply rights the state related data to the cookie, and with every new request, examines the cookie content and use the state information therein to further process the request based on the previous state. The server sends a “Set-Cookie” header (Figure

4) to the browser with each HTTP cookie, and it is browser's task to store it within browser's cookie store. During subsequent resource requests from the server, the client browser first checks its cookie store for a non-expired cookie belonging to the server domain. If found, that cookie will be returned to the server with a "Cookie" request header (Figure 3). As the cookie exchange process between the client and server is a part of the protocol (Kristol & Montulli, 1997), neither the server-side nor client-side applications need to implement additional mechanisms to send and receive *state data*.

Though HTTP cookie was introduced as a state management mechanism (Kristol & Montulli, 1997), it was soon discovered that it can be also used for tracking user activity, across domains (Kristol, 2001). For the purpose of this research other state management mechanisms discussed above, are not considered as they cannot be used for cross-domain tracking, but only for state management within the local domain.

HTTP cookies

The original IETF cookie specification of the RFC 2109 (Kristol & Montulli, 1997) has since been twice replaced, most recent being the current RFC 6265 in 2011 (Barth & Berkeley, 2011). HTTP cookies are important and integral part of this study, as they have been the *de facto* tracking mechanism, which this study seeks to improve on. The specification requires that browsers implement at least 4096 bytes per cookie, 50 cookies per domain and at least 3000 cookies in total, within browser's cookie storage. Cookies have following attributes (Table 3) that can be set by the originating webserver. Each cookie is assigned a name and a value, in addition to a collection of key-value pairs that can store information (Barth & Berkeley, 2011).

Table 3: HTTP cookie attributes

Attribute	Description
Name	The name assigned for the cookie. If no name assigned, the <code>Set-Cookie</code> command will be ignored cookie will and not be set.
Value	A single value or a collection of key-value pairs
Expires	A date and time the cookie will expire. If a server wants to remove an existing cookie, it sends a new cookie with the same <i>Name</i> , <i>Domain</i> and <i>Path</i> , but with a date in the past, which will cause the browser to remove the cookie. If expiry is not set, it essentially is treated as a <i>Session cookie</i> which will be discarded at the end of the current session. This attribute was introduced in latest specification RF 6265, making it easier to set the expiry date than the <i>Max-Age</i> attribute of the two previous specifications RF 2109 and RF 2965.
Max-Age	Number of seconds, the cookie should remain valid. If <i>Expires</i> attribute and <i>Max-Age</i> both are set, <i>Max-Age</i> takes the precedence.
Domain	If left blank, it assumes the current domain of the webserver that sent the cookie, but not used for any sub-domains. The domain name cannot be for a different domain or any Top-Level Domains (TLD) such as <i>.com</i> , <i>.net</i> or <i>.co.nz</i> . This attribute's set scope should include the domain of the sent server. When this attribute is set the cookie will be sent to the domain of origin and any subdomains that is part of this domain. The original specification RFC 2109 required a leading dot in the domain name attribute, but not required by the latest specification, therefore ignored if present. But a trailing dot invalidates the attribute and causes to ignore this attribute completely (Barth & Berkeley, 2011).
Path	If not set, assumes the current directory path of the requested resource on the server. If specified, the cookie will be sent on any resource requests to the specified path or its subdirectories.
Secure	This attribute limits the scope of the cookie to only secure connections, defined by the browser, usually using HTTPS protocol
HttpOnly	This attribute limits the scope and access to HTTP requests only, thus restricting access to scripts.

If the client-browser receives a new cookie with the same cookie-name, domain, and path values as of an existing cookie, in its cookie-store, the existing cookie will be replaced with the new cookie. The webserver can choose to remove the cookie stored in browser's cookie-store entirely, without replacing with a new cookie, by sending a new cookie with the *Expires* attribute set to a date in the past.

It also allowed users to log-in to a website, and browse from page to page, while remaining logged-in, as "remembering" state was possible with the use of HTTP cookies. Such HTTP cookies set by the website being visited by the user, is called *first-party* cookies. However, some webpages are composed of resources from multiple external websites: Such webpages may contain images, CSS stylesheets, JavaScript files, audio and video files, widgets that display currency rates, weather, visitor counters, etc. from other external *third-party* websites. During the page loading process, when the browser makes requests for those resources from the third-party websites, those external sites may also choose to place their own HTTP cookies in the user's browser. Such cookies are called *third-party* cookies. The webserver returns the requested resource together with a "Set-Cookie" HTTP header that causes the client browser to save that HTTP cookie in its cookie-cache. Even a JavaScript can cause an HTTP cookie to be set by invoking the "document.cookie" function. An important characteristic of the HTTP cookie is, that the browser will always return the cookie back to the same webserver that originally set the cookie in client browser. For instance, if a user requests a webpage from example.com, the webpage is accompanied by a cookie that originated at example.com. The webpage also has links to two resources: one from external1.com and another from external2.com. The user's browser then sends a request each to external1.com and external2.com, and receives each resource accompanied by a cookie from each domain.

The browser stores all three cookies in browser's cookie-cache. On the next request to example.com, either during the same browsing session or at a later date, the browser will always return the cookie from example.com, that it received previously, but it does not return the two cookies from

external1.com and from external2.com, to the example.com server, as the cookies are only returned to the originating webserver. But at any later stage, should the browser make a request from external1.com or external2.com servers, the respective cookie will be returned. If the user visits other websites at a later date, that also have resources from external1.com and external2.com embedded in their webpages, during the page load process the browser will send HTTP requests to the external1.com and external2.com, each accompanied by external1.com and external2.com cookies that were placed originally when visiting example.com webpage.

This is an important behaviour of the cookies in a browser, that enables using cookies for cross domain tracking. Such tracking service needs to embed a shared resource in as many different websites as they can, so that any visitors to those websites, can be tracked by the tracking service. If the webserver stored a unique identifier for each user in a cookie on the user's computer, during subsequent requests, the tracking server will be able to identify the user by reading the unique identifier, which is the process of tracking a user over time. Though HTTP cookies were not meant to provide tracking capabilities across domains, the provided state management capability has made cross-domain tracking possible (Kristol, 2001).

Though a user can disable HTTP cookie usage within the browsers, which was becoming a popular security option a decade ago, the benefits of using HTTP cookies outweigh the security risks, in the "connected" world of today. Most browsers come with cookie-enabled by default, though a user can disable this option after navigating through a not-so user-friendly menu system. Even when persistent cookies are disabled, the browsers still use "Session-cookies" which offer same capabilities as persistent cookies, except that they are not saved on to the hard disk and therefore are effectively available only for the duration of the current browsing session. Yet, the tracking process during the specific browsing session still takes place, and in scenarios such as AM models, the commission earnings for the affiliates who caused visitors to visit some e-commerce sites, can still take place (Amarasekara & Mathrani, 2016).

Though there have been some rumours and speculations a few decades ago, that cookies can harm computers and they can scan hard disks to steal passwords, credit card numbers, etc. cookies do not have executable code. Cookies are stored in plain text format, that cannot harm a computer, unlike malware (Harding et al., 2001). It is now widely accepted that cookies do not pose any significant risk, instead they can be used to enhance user experience (Kristol, 2001). Securely encrypted HTTPS connections are used to connect browsers with web servers, hindering Man-in-the-middle attacks. Sivakorn et al. (2016) explores the dangers of mixing HTTP and HTTPS during a session, which can lead to cookie-hijacking. Browsers save cookies on the hard disk of the local computer either as text files or in a database.

If a computer has more than one browser installed, the browsers do not share their cookies between them (Logan & Mossing, 2007), therefore access to a server from two different browsers on the same computer will appear as two different users to the server. This is a significant issue for tracking technologies, that a robust tracking system need to try to solve. In the AM section below, what implications it has in e-marketing strategies is discuss further. The *super-cookie* concept and *Flash cookies* (section 2.6.1) claim to address this issue efficiently and effectively.

During the browsing session between a client-browser and a server, though the originating client-IP address is visible to the server, it is not a unique way to identify individual users or computers on the Internet (Xie et al., 2007). If there is a “proxy server” between the user’s computer and the web server, each request to the server will have the same originating IP address, which is the IP address of the proxy server, not of the individual user’s computer. Large corporate networks or Internet Service providers (ISPs) use proxy servers to cache content, to reduce network traffic (Kristol, 2001, p. 6). Even at a home network, most routers will use Network Address Translation (NAT) routing, which hides the individual IP addresses of computers within a home network environment. The web server sees only the public IP address of the NAT router.

A cookie with a unique identifier sent by the server to the client browser on the initial contact, is the best way to identify a client-browser uniquely. In this case, “Identify uniquely” does not mean that the server can identify an individual uniquely, but it can identify a specific client-browser only (Kristol, 2001, p. 7). In domestic scenarios, where many family members share a computer, with only one user profile within the operating system, any family member using a specific browser, for instance when everybody is using the Google Chrome browser, the cookie stored in the Chrome browser will identify every family member with the same cookie identifier, which makes the webservers that are accessed, assume it is the same person behind the computer and browser. Conversely, if there are more than one browser installed on the computer and if one specific family member is using Chrome browser and Firefox browser at different occasions, a server will identify that person as two different people, as each browser will have a cookie with a different identifier. Different browsers on the same physical machine do not share cookies. That makes for instance a family of five people, to be identified as one person. In an alternative scenario, if one person has five browsers installed on one machine, and used all five browsers interchangeably, then that one person appears as five different people to the webserver. To achieve a better result in identifying individual users, additional techniques can be used, beyond the cookie identifier. If the user chooses to share the user’s name, address, or other personally identifiable information with the server, at the time of opening an account or a user profile, then the server can save that information together with the unique identifier, which only then allows the server to personally identify a user. Else, accessing the public identifier of a OSN account, can uniquely identify a user, even in a shared family computer scenario, which we will discuss further in the discussion section. Public IT infrastructure such as those computers in a public “Cyber-café” or at a backpacker hostel, where large numbers of people access popular websites from one computer can cause a dilemma, trying to identify a user uniquely. This issue is fast becoming less relevant than a decade or two ago, with personal mobile infrastructure such as smart phones, tablets and laptops being increasingly used by one individual user only. It has created a connected world with individual identities on the Internet becoming the norm, keeping people connected with their own individual

OSN and email accounts. That has necessitated families who might use a shared computer to create user profiles at operating system level, to keep their unique identities even in a shared computer. Re-visiting the scenario discussed earlier, browser cookies are not shared among user profiles, hence each user in a family unit, when logged in to their own user profiles, even if they use the same browser, the browser within each profile will have a separate cookie, that can identify individual users.

2.3 Tracking on Internet

Different entities track user activity for various purposes. Some of the cases are:

- i. Websites want to remember customisations and personalisation made by visitors during their previous visits, to offer an improved user-experience.
- ii. Online advertisers, search engines and other e-marketers attempt to personalise advertisements based on a visitor's historical browsing data (Libert, 2015). Without this capability, Internet users can feel hassled, when products and services that do not even vaguely interest them, appear at most of the websites they visit (Hoofnagle et al., 2012). Also, the advertisers will be wasting their marketing budget on audiences that do not yield them any positive outcomes.
- iii. AM model, which is one of the most cost-efficient online marketing methods available to e-marketing practitioners. It needs the capability to track visitors who are viewing and clicking on advertisements placed on affiliates' websites (Brear & Barnes, 2008; Norouzi, 2017; Pawan & Gursimranjit, 2020; Suryanarayana et al., 2019). The tracking mechanism traces clicks and successful outcomes; and pays commissions to affiliates.
- iv. Customer behavioural data within an e-commerce site (e.g., duration spent on site and on specific pages, products perused, success rate, etc.) are useful for a marketer, and can be easily generated within the e-commerce application. By subscribing to an external business analytics provider, such data can be combined with customer demographics obtained through insights over interactions beyond the boundaries of the practitioner, to generate richer

person-profiles useful for a marketer (Baumann et al., 2019; Castelluccia, 2012; Dwyer, 2009).

- v. Security establishments use tracking technology to identify people who are deemed a security threat. They are flagged across multitude of websites and their activities are monitored (Englehardt et al., 2015).
- vi. Third party companies such as Cambridge Analytica profiles people with the help of people's social media affiliations and interests. By using such profiling methods, they are capable of undertaking nefarious activities such as influencing and creating biased opinions to manipulate political and election outcomes in many countries around the globe (Bakir, 2020; Manokha, 2018; Margaret, 2020; Richterich, 2018).

Case (i) involves managing state within a single domain and therefore do not fall in to cross-domain tracking that we under this study. Cases (ii), (iii), (iv) and (vi) involve cross-domain tracking within an e-commerce context, which directly aligns with our study. Case (v) is similarly a case of cross-domain tracking, but not within an e-commerce context, therefore is not considered in this study. Nevertheless, many tracking related issues that we discuss are applicable to the case (v) too and our findings can be useful for research activity in that area. Cases (ii) and (iii) involve Internet traffic generation strategies. As every e-commerce site needs to attract customers to their sites, different traffic generation strategies that we discuss in the following sub-sections involve spending large online advertising budgets. Our research outputs are aimed at improving the reliability of the underlying technology, which we will discuss throughout the next chapters. Cases (iv) and (vi) are business models related to generating business insights and digital personas, that are very privacy intrusive. Case (iv) is useful for marketers for target marketing and customer segmentations. Business analytics are vital for business managers to make informed decisions. In some instances, such business insights are derived within the processes discussed in cases (ii) and (iii). Case (vi) represent operators at the top of the tracking hierarchy, such as in case of OSN and large scale business analytics providers (Richterich,

2018), whose purpose for tracking is not merely as an underlying technology to accomplish e-commerce operations or gathering business insights on their own customer-base to improve interaction with customer, but to market personal information as a commodity (Bakir, 2020; Laterza, 2018; Margaret, 2020; Richterich, 2018; ur Rehman, 2019). It has become a contentious issue that is attracting legal implications, which can negatively impact all tracking technologies and instances in general, including those under cases (ii) and (ii), which are usually not privacy invasive, but a technical necessity. Therefore, this study will include cases (iv) and (vi), to describe the levels of privacy intrusions associated with different practices.

2.3.1 Affiliate marketing model

Digital platforms built over digital infrastructures allow multiple stakeholders to orchestrate various online services across distributed resources and participants (Constantinides et al., 2018). E-marketers are on the constant lookout for ways to generate visitor traffic to their e-commerce sites in a cost-effective manner. Search Engine Optimised (SEO) page rankings, search engine advertising, keyword bidding, CPM display advertising and CPC banner advertising are some of the different ways to attract user traffic for a fee. With the advent of Affiliate Marketing (AM) businesses around the globe found a new way to generate visitor traffic at a relatively low cost, using a network of affiliates (Brear & Barnes, 2008; Norouzi, 2017).

AM platforms provide an easy-to-access unified ecosystem that binds external parties and facilitates connections between supply-and-demand scenarios. It defines a way to generate visitor traffic to an e-commerce site, through a network of independent websites called affiliates, against a fee or commission. Affiliates represent influential partners who endorse some product, service or brand with the intention to influence consumer purchase decisions by using electronic word-of-mouth promotion strategies (Ismagilova et al., 2020). Affiliates are typically in the limelight, having the reach of the intended customer segment; therefore, they undertake promotion of those e-businesses that have been ratified by them. Affiliates rely upon the information-seeking behaviours of visitors to their own

website. The affiliate's expectation is that a passive display of an e-business advertisement that appears on the affiliate's website might catch the attention of some visitor, who might then click on it. For example, a travel blogger who writes an online travel journal on their recent travel expeditions in New Zealand would have the reach of prospective travellers to New Zealand; therefore, they are an ideal affiliate to promote tourism-related services such as accommodations, flights, rental cars, or adventure activities. On the other hand, this travel blogger would not be the best affiliate to promote a service such as real-estate sales in London. Hence, advertisers try to find affiliate websites that carry content related to their products. Similarly, affiliates too look for advertisers of products that relate to their web content.

E-commerce has enabled business enterprises to reach customers around the globe far beyond the geographical boundaries and it has opened up opportunities for SMEs to reach markets that were only accessible to multi-national conglomerates, before (Amit & Zott, 2001; Gregori et al., 2013; Mariussen et al., 2010). E-commerce sites follow different strategies to promote visitor traffic to their sites, hoping that some of the visitors might make a purchase. Search engine visibility is the starting point. Paid advertising, keyword-bidding, re-marketing strategies are very efficient, but need technical expertise (Rutz & Bucklin, 2007), which is not readily available to SMEs. AM has filled the gap in online marketing strategies as a lifeline for SMEs (Dennis & Duffy, 2005). It is equally popular among the largest players such as E-Bay and Amazon.com.

This has boosted opportunities for SMEs as they gain global visibility and can reach markets that were earlier accessible only to multinational conglomerates (Gregori et al., 2013; Kilubi, 2015; Mariussen et al., 2010). An attractive and professional-looking website of a home-grown business can appear as a large enterprise to customers. But first, businesses need to identify innovative ways to generate visitor traffic to their websites. Search-engine visibility is the starting point. While paid advertising, keyword-bidding, and remarketing strategies are efficient online marketing strategies, they require technological expertise, which is not easily available to SMEs.

AM has filled this gap to provide a lifeline for e-businesses; it works toward providing information channels that are aimed to increase online sales by showcasing and distributing products more efficiently.

Wilson (1999) contends that four possibilities are prevalent in information-seeking behaviours, namely, passive attention, passive search, active search, and ongoing search. For instance, a consumer may watch an advertisement of a product without any intention to act on it (passive attention), but it could lead to casual browsing (passive search) or a more meaningful information search (active search) that in turn stirs more interest such that the consumer may continue their search to get more product details (e.g., price range, user ratings). Therefore, AM builds on the “information needs” of consumers by using innovative digital interventions to facilitate online sales in a cost-effective manner. Emerging disruptive technologies are challenging the status quo in the management of e-business operations as new ideas are being translated with ongoing technological advancements to capture customers globally (Broekhuizen et al., 2018).

Enhanced browsing tools for hyper connecting consumers with e-businesses are spread across distributed platforms. At the same time, however, these pervasive technologies used for profitability purposes (e.g., reduced costs, fast delivery times) come at a price since they also increase our digital vulnerability (Ransbotham et al., 2016). In the pursuit of optimising pricing structures by increasing automation and resource efficiencies, we may have opened a panacea of unknown possibilities, including that of vulnerabilities. For instance, the CPA method earlier appeared as a disruptive technology and was considered immune to numerous frauds that were prevalent in other AM advertising models. However, the litigation against Shawn Hogan brought to attention that it is not as safe as was envisaged (Edelman & Brandi, 2015). Nevertheless, the CPA method is still considerably safer and more cost-efficient than other visitor-traffic-

generation methods, although better robust security measures to safeguard this most cost-effective marketing model are further needed.

Stakeholders in AM

AM model comprises of four main actors as shown in Table 4: An “Advertiser”, who is an e-commerce site who wants to generate visitor traffic, with the hope of selling the products to some of those visitors. Apart from visitor-traffic based on organic search results, they depend on paid-traffic generation models.

An “Affiliate”, which is an independent website, that has a good reach of possible customers. The affiliate is willing to forward such visitors to the e-commerce site, on the basis, that the affiliate receives a monetary reward for doing so. The above e-commerce site would have many such affiliates who are forwarding visiting traffic the e-commerce site, hence there needs to be a mechanism to track and monitor the visitor traffic; also track the visits that convert to a monetary outcome and calculate the payments to the individual affiliates based on the traffic that they forwarded. Though larger e-commerce practitioners such as e-bay does this in-house, many small-scale practitioners do not have the technical expertise, hence subscribe to a third-party AMP who does this on their behalf.

The fourth actor is the “Visitor” to the e-commerce site, who is a potential customer who may fulfil a monetary outcome at the e-commerce site of the advertiser, such as buying goods, signing-up for a membership or subscribing to a service, etc. (Amarasekara & Mathrani, 2015).

Visitor traffic generated in the affiliate’s website ends in the advertiser’s e-commerce site, although both parties are dispersed geographically; additionally, their websites are hosted in different domains with different web infrastructure. A visitor might make a purchase on the first instance of arriving at the advertiser’s e-commerce site or might choose to return a few days later to complete the purchase. Such temporal separation between exposure to an advertisement and subsequent action by the visitor is indeed common (Asdemir et al., 2012). The visitor’s intention to purchase on a later date is a result

Table 4: Stakeholders in Affiliate Marketing

Actor	Role
Advertiser/ E-business	E-commerce practitioners who advertise their merchandise aiming to generate visitor traffic to their website with the anticipation that some of those visitors will purchase their products.
Affiliate/ Social Influencer	An independent website (or a social intermediary) with many online readers (or visiting traffic). Affiliates have a good reach of possible customers who might be interested in the merchandise sold by the e-commerce website (advertiser); therefore, they can refer or forward their readers to the e-commerce site. Referrals are made on the basis that the affiliate receives a monetary reward for doing so.
Affiliate Management Network (AMN)	A third-party technology platform that provides services to advertisers and affiliates, since each advertiser has many affiliates who are forwarding them visitor traffic. The platform tracks and monitors visitor traffic, keeps record of visits that convert to a monetary outcome, and calculates commissions earned by individual affiliates based on the traffic they have forwarded. In the case of very large e-commerce practitioners such as eBay and Amazon, the advertisers themselves carry out this function in-house.
Visitor/ Potential Customer	A potential customer of the e-commerce site who may fulfil a monetary outcome, such as buying goods, signing-up for a membership, or subscribing to a service from the advertiser. The visitor becomes a consumer on purchasing a product or service from the e-commerce website.

of the affiliate's influence; therefore, the affiliate deserves a commission from the e-commerce site. Hence, e-marketing efforts require a tracking system that can track each visitor's information-searching and information-using behaviour reliably and accurately, across multiple domains and over a period of weeks or months, as determined by the e-marketer's business policy. The tracking technology provider (or AMN) places an HTTP cookie on the visitor's computer to achieve the above outcome.

AM traffic generation models

Under AM model, an e-commerce site can use different traffic generation models (Table 5) such as Cost-per-mille (CPM), Pay-per-click (PPC) which is also known as Cost-per-click (CPC), or Cost-per-Acquisition (CPA). Under CPM the advertiser pays an agreed fee to the publisher for displaying 1000 advertisements to potential customers. CPM is the cheapest, costing around a

Table 5: Visitor-traffic Generation Schemes, Pricing Structure, and ISB in an AM ecosystem

Traffic-gen. scheme	Pricing structure	Information-Seeking Behavioural outcome
Cost-per-mille (CPM)	The advertiser pays an agreed fee for displaying 1000 advertisements to visitors (or potential customers).	Passive attention (fleeting thought)
Cost-per-click (CPC) or Pay-per-click (PPC)	The advertiser pays for each visitor who clicks an advertisement link or banner in the affiliate's webpage and forwards the visitor to the advertiser's e-commerce site.	Passive search (casual browsing) or active search (active scrutiny)
Cost-per-acquisition (CPA)	The advertiser pays an agreed sum or, more frequently, an agreed percentage of the total sales value as a commission to the affiliate only when a visitor makes a purchase at the e-commerce site.	Search resulting in intervention (search information is used to make a purchase)

few tenth of a cent to display one advertisement, as there is only a very small chance of one of the visitors who saw the advertisement might indeed click it and follow through to make a purchase at the e-commerce site (Faou et al., 2016). Only about 1% visitors to an affiliate website actually clicks on a banner advertisement (Benediktova & Nevosad, 2008). Under CPC, an advertiser pays a higher fee than CPM, in the range of a dollar for each visitor who clicks and advertisement link or banner in the affiliate's webpage, which forwards the visitor to the advertiser's e-commerce site (Faou et al., 2016). With both of these methods the advertiser might lose a lot of advertising budget paying for unintended traffic of visitors, who wouldn't buy a product at the advertiser's e-commerce site (Kayalvizhi et al., 2018). To target the appropriate market segment of potential customers, advertisers

need the technical expertise of e-marketing, just as in other e-marketing endeavours such as Google Ad-words, which most SMEs lack. CPA is by far the most popular marketing model, without the risk of losing an entire marketing budget overnight, unlike other e-marketing technologies (Dennis & Duffy, 2005). It also shifts the traditional marketing paradigm: Instead of the need to spend an advertising budget beforehand, without any guarantee of returned benefits, under CPA advertising model cost for traffic generation is only paid to affiliates after a sale has occurred, thus guaranteeing a return on advertising cost, and not needing an advertising budget beforehand (Norouzi, 2017).

Though not as prevalent as click-fraud in CPC model, recent research has shown that CPA is not the silver bullet that solves all the e-marketing problems for SMEs, as earlier perceived. Litigation against Shawn Hogan (Edelman, 2015) who fraudulently collected commission worth over 28 million dollars from eBay in 2014, have drawn attention to the fact that even CPA is not immune to fraud activities. Chachra et al. (2015) have established that such fraud is seemingly marginal presently within the AM environment but has the potential to become widespread in the future, if the vulnerabilities that are being detected currently, are not addressed now. Contrary to these findings Snyder and Kanich (2015) who used a different methodology to assess the volume of fraudulent activities found 38.1% of the click traffic was fraudulently generated. Our own previous research (Amarasekara & Mathrani, 2017) found considerable number of fraudulent activities among click-traffic dataset.

Trust is an important factor, among all the stake holders associate with the AM value chain (McKnight et al., 2002). In most situations the advertisers have not met or know the affiliates personally and have a very limited knowledge of each other's businesses and reputations. Often the two parties agree to abide by a set of rules defined by the advertiser and these rules can be very different between advertisers. Some could have exactly opposite rules than the others. For example: Some advertisers, who use other forms of e-marketing available to them such as Google AdWords, could prohibit "keyword bidding" or "typo-squatting", as they do not want to have their own affiliates competing for

the same keywords. Another advertiser, who does not use Google AdWords, would encourage an affiliate to use keyword bidding, as that would increase traffic generation from multiple sources.

Tracking process in AM

There are two separate tracking processes involved in an AM system. “Click tracking” is the process of tracking the visitor’s click-action at the affiliate’s website, followed through to the arrival of the said visitor at the advertiser’s e-commerce site. “Conversion tracking” takes place, when such a visit “converts” to a desired outcome, such as buying a product, or signing up for membership or any other expected outcome, the said outcome is tracked and recorded.

Figure 5 provides a logical view of the AM process, starting from a visitor’s click on a banner advertisement at an affiliate’s website to the completion of a purchase action. The sequence of the processes involved are numbered in the diagram. When a user views an affiliate website (process 1) and clicks an advertisement link (process 2) the “Click Pixel” embedded in the webpage causes the tracking server to create a record of the “click” action in the database (process 3). The tracking server then sends a cookie to the browser with a unique identifier that refers to this specific click. It also sends a redirect response to the browser, targeted at the advertiser’s e-commerce site (process 4). The visitor then browses the e-commerce site and makes a purchase decision (process 5). The process 6 is abbreviated as “AN Res. Rq.”, which stands for “Affiliate Marketing Network resource request”, which refers to the “Conversion Pixel” embedded in the payment confirmation page sent by the e-commerce server. In the background without any visible clue to the user the Conversion Pixel causes the user’s browser to send a resource request to the tracking server with the information such as the Invoice Identifier, total purchase price, etc. as parameters of the resource request. As every HTTP request to the web server is accompanied by the cookies that the server has set previously, in this case during the click-tracking process numbered 3, the tracking server records the sales conversion details against the click-tracking data in the database. The Click Pixel and Conversion Pixel are small pieces of JavaScript code that are embedded in those webpages that provide user-specific information

to the tracking server. The tracking server, while being invisible to the user, keeps track of all processes and traffic movements with the help of tracking-cookies.

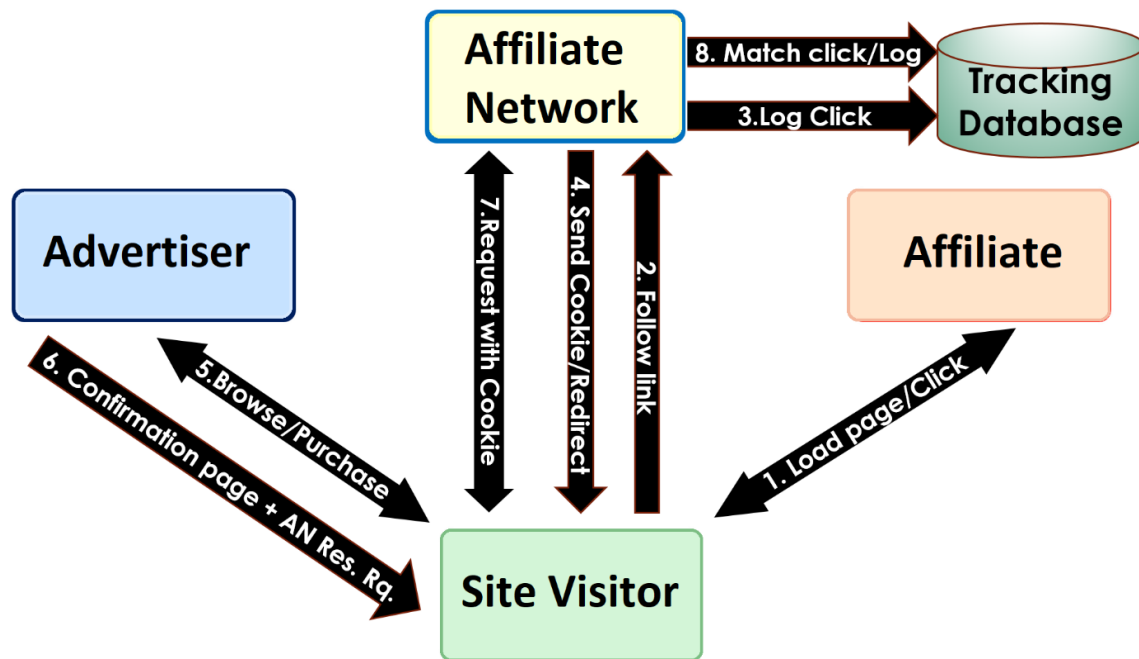


Figure 5: Tracking process of the Affiliate Marketing Model

Click Tracking

The homepage of an affiliate usually contains a “click-Pixel”, which is a banner advertisement image or a hyperlink within the text description. The hyperlink points at a click-tracking URL on the tracking server. Data that need to be passed to the tracking process such as affiliate and advertiser identifiers, offer ID etc., are passed as URL parameters (Amarasekara & Mathrani, 2017).

When a site visitor clicks on the banner advertisement or on the text hyperlink, visitor’s browser sends a resource request to the click tracking URL. The tracking server then logs the parameter data into a database and redirects the request to the advertiser’s e-commerce site based on advertiser identifier on the request parameter. With the response to redirect, the tracking server also sends a HTTP cookie to the visitor’s browser, which the browser will add to its cookie collection. During any future interactions with this domain, the browser will always send this HTTP cookie (Amarasekara &

Mathrani, 2017). Though some of the previous researchers found overwriting cookies during the above process to be one of the more seriously adverse effects of affiliate frauds such as cookie stuffing (Chachra, Savage, & Voelker, 2015; Edelman & Brandi, 2015; Snyder & Kanich, 2015), our research shows that how the HTTP cookies are handled is purely a business decision of the advertiser, which depends on how the commission is shared among multiple affiliates. The business rule dictates if the last affiliate gets all or if the commission is shared among all the affiliates who contributed. Accordingly, either the cookie can be overwritten, or each affiliate identifier can be added to the affiliate list on the cookie (Amarasekara & Mathrani, 2017).

Conversion Tracking

A “*Conversion-Pixel*” is embedded in the transaction confirmation page of the advertiser. A *conversion-Pixel* is usually a hidden HTML element with its source pointing at the conversion tracking URL on the tracking server. The data fields such as advertiser’s identifier, transaction identifier, total price, which are needed for the tracking purpose being passed in the URL as parameters.

When a visitor completes a payment transaction, the transaction confirmation page with the embedded hidden *conversion-Pixel* causes the conversion tracking server to log this information in to the conversions database. As every purchase of every visitor to the e-commerce site, direct customers and those that arrived via an affiliate site, gets a confirmation page with the hidden iframe, every sale will be notified to the tracking server. If the resource request to the tracking server is not accompanied with a HTTP cookie, it indicates a direct sale. The conversion tracking process examines the affiliate ID or IDs mentioned in the cookie and finds the corresponding click record and reconciles the two records with the pre-agreed percentage of the commission calculated against the total transaction amount and paid to the affiliate (Amarasekara & Mathrani, 2017).

Fraud and vulnerabilities in AM

CPM and CPC advertising models do not guarantee a return on the investment of the marketing budget and are prone to numerous fraud scenarios including click fraud (Hu et al., 2013). CPA was

considered the silver-bullet that solved the risk of affiliate fraud in AM, as the affiliate earns a fee or commission only when a visitor makes a purchase at the advertiser's e-commerce site. But the "cookie-stuffing" fraud discussed by Edelman & Brandi (2015) shows that even CPA marketing model is not immune to fraud. Since then, my master's degree research work has uncovered further vulnerabilities facing AM model, which are presented below.

Cookie stuffing

This fraud involves placing a HTTP cookie or many cookies from many different advertisers into a visitor's browser, without the visitor having clicked any advertisement. A legitimate affiliate could carry more than one banner advertisement from different advertisers, but often too many banners on one site can put off site visitors and the credibility of the site can be at risk. With Cookie stuffing method, a rogue affiliate can keep the webpage free of advertisements but maximise profit by stuffing as many cookies in the background. For example, if the affiliate site is a travel blog about travelling around in New Zealand, it is highly likely that many visitors might be planning a visit to New Zealand sometime soon. The affiliate would stuff cookies from as many cookies from each hotel chain, each car rental company, Airlines, tourism related activity providers and of places of visits, as it is likely that the site visitor might visit some of those websites and book a product or service. At such time, the previously stuffed cookie will identify the rogue affiliate as the source of the visitor traffic and will pay him a commission. This allows a rogue affiliate to cash in large amount of money from multiple AM practitioners using one site visitor. How can cookies be stuffed into a browser? A rogue affiliate can use a technique such as "load-time click (Amarasekara & Mathrani, 2017).

Load-time click

While displaying an advertisement free webpage, the affiliate can have a JavaScript code segment that runs at page's load event, which sends a resource request to each of the tracking server of each advertiser, which should legitimately only run on a click action of the user. As the visitor browses the webpage, unaware to visitor, the visitor will be "clicking" on large number of advertisements. Using

JavaScript Load method can be used within CPM or CPC scenarios, where just a request sent to the tracking server triggers the required result, but not in CPA scenarios, as Cross Origin Resource Sharing (CORS) restrictions on cross-site scripting (XSS) prevents browser accepting a cookie, which in turn fails the tracking process. Therefore, in a CPA scenario, a rogue affiliate would use an embedded hidden *"iframe"*, a resource request within the HTML code for a hidden element or within a CSS file (Amarasekara & Mathrani, 2017).

Conversion hijacking

Instead of stuffing cookies in browsers of thousands of site visitors hoping that some of them might really visit one of those e-commerce sites and make a purchase, a rogue affiliate can do a conversion hijacking, by dropping a cookie in to a direct customer's browser, just before completion of a purchase. This can be achieved using Adware (Edelman & Brandi, 2015) or similar malicious software installed on a user's computer. This can also be a part of an internal threat from the e-commerce site's perspective, as we discovered in our investigation, where a rogue affiliate gets the cooperation of an employee with sufficient privileges to embed a small piece of code segment on the Web Server.

A similar result can be obtained, without having to do the above in real-time, during the purchasing action of a visitor, but by triggering the conversion *Pixel* code that is embedded in the e-commerce web server, with the appropriate parameters. The solutions that we propose further below will help minimise multiple variations of this threat.

From the advertiser's perspective, some threats originate from external sources such as, by affiliates, site visitors and hackers, etc., 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 we established and subsequently tested using our prototype, is what we call "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 AM, “conversion stealing” can lead to large losses for an advertiser. This risk can originate internally or externally, though it is very much 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 tracking entries, in fact refer to legitimate transactions within the advertiser’s system (Amarasekara & Mathrani, 2017).

Conversion Stealing

Broad group of frauds fall into this category. Apart from AM, e-commerce practitioners use other traffic generation methods, such as paid searches, paid advertisements or visitor arrive through organic searches, which are unpaid searches through search engines. As all online transactions trigger the conversion *Pixel* in the confirmation page, therefore all transactions get recorded in the tracking database, those traffic that was not generated by CPA model of AM, is marked as non-commission paying transactions. A rogue affiliate can use one of the few possible methods to claim such transactions to affiliate’s account, which is “Conversion Stealing”. One of the most serious of frauds would be updating tracking database en masse, assigning an affiliate ID against selected non-commission earning transactions, which would cause the affiliate to receive those unpaid commissions. Even a reconcile application would successfully reconcile such transactions, and would be hard to track, unless many additional checks are done, creating processes that are specifically targeting this fraud. Usually, this would be an internal fraud, where an employee of an e-commerce site, third-party support staff or employees of the AMN with sufficient privileges is in a position to create an automated process to filter individual non-commission paying transactions and use web APIs to update the tracking server. An external player can also do this using web API’s if the fraudster has access to Web APIs, else trigger the same conversion *Pixel* code that is embedded in the confirmation page of the e-commerce site, with the correct parameters. An internal staff can find the parameters

easily through the transaction database, but an external fraudster will need to find parameters often through brute force or with trial-and-error guessing. During our case-study, we found evidence of brute force attacks, and a few other methods to find transaction IDs. One was to make a booking or a purchase and guess the subsequent transaction numbers based on the booking's transaction number. In a reservation site, we noticed fraudsters have entered those guessed numbers into the reservation retrieval service, which allows a reservation to be retrieved by customers to change their reservations. Hence, we have proposed a reservation buffer system in our recommendations section below (Amarasekara & Mathrani, 2017).

Typo-squatting

Some of the frauds such as typo squatting and keyword bidding are considered a fraud by some advertisers, while others consider it legitimate. That depends on the contractual agreements and different marketing strategies used by the advertiser. Typo squatting is when an affiliate acquires domain names that are very similar to an advertiser's domain and captures the traffic of visitors who either mistypes the advertiser's name or types a confusingly similar name (Edelman & Brandi, 2015). After capturing the visitor, the affiliate can either redirect to the intended advertiser with a cookie to identify the affiliate thereby earning the commission or forward to different rogue website.

All the above fraudulent actions need certain amount of technical expertise, as they are technology based. There are other non-technology based frauds that too can incur heavy losses using simple deceitful actions. On most e-commerce sites, for example at a hotel or a car rental company an affiliate can book a car or a room himself, a few months ahead and earn a hefty commission before eventually cancelling the booking after a few months. Many advertisers pay the commission at the end of the month following the purchase. The AMP integration application that we examined for the purpose of this research was capable of effectively controlling this category of fraud, by reconciling AMP conversion records with the back-office databases of the advertiser. Credit card frauds, "click factories" where large numbers of staff are hired to manually click on advertisements, in countries

where labour can be found cheap and using posts in OSN as “Click-bait” are some of the more manual frauds.

Checking the referrer HTTP header is useful to detect typo squatting fraud to some extent, but many typo-squatted domains use redirection chains (Chachra, Savage, & Voelker, 2015; Vacha, Saikat, & Yin, 2013) to avoid detection (Amarasekara & Mathrani, 2017).

2.3.2 Business Analytics

Business managers in general, and marketing teams more specifically, depend on business analytics for their enterprises level strategic planning. Business analytics provide insights on Customer demographics, buying habits, marketing campaign impact, visitor counts, etc. and enable marketers to create targeted campaigns for customer segments for better outcomes. Every e-commerce site has a product catalogue, shopping cart facility and some form of a transaction processing facility. Many software contains capability to track a user’s browsing behaviour from the time they arrive at a landing page, throughout their browsing sessions including purchasing actions. This empowers marketing teams to evaluate the performance of specific marketing campaigns, fine-tune the visitor traffic generation models such as the search engine advertising plans or AM campaigns based on higher commission rewards etc.

Marketers also are interested on what landing pages did the customers arrive at, which affiliates promoted that traffic, which products were perused by the visitors, and how long they spent on which products, which pages, and products were skipped, showing which products are in demand. Finally, if any products were purchased and if so, what other products were bought in combination. If no purchase were made, then insights, what the reason would have been, and ideally, where did they go to from this e-commerce site. Business analytics generated through tracking data provide helpful insights for a marketer to understand if the customer needs are met by their product offerings. While many e-commerce software products have the features to gather business analytics, often business

marketing managers want to tweak their marketing campaigns based on the results, so that they can further tweak them to achieve optimum outcomes. As they often don't have access to change any code within the e-commerce software, and often it takes a long development cycle to implement such additional changes, they tend to gravitate towards external partners who offer business analytics for free or a more premium services at a reasonably low fee, such as Google's Universal analytics (Castelluccia, 2012; Dwyer, 2009; Roosendaal, 2012).

If all the information gathering takes place within one domain, the domain of the e-commerce site in question, first-party cookies can be used to track the user-interaction. Else other non-cookie based state management techniques discussed above, such as embedding into the request URL, hidden form field with the body of the page, etc. Though new privacy regulations such as General Data Protection Regulation (GDPR) allows cookies that are functionally necessary for a website (GDPR, 2016), such customer demographics gathering would not fall into the functionally necessary category. The privacy breach becomes even more concerning, then the tracking is entrusted to a third-party tracking provider, as the customer information leaks out to an external domain that was never visited by the user. This research studied the extent of the privacy breach in this situation and how much of privacy information can be combined in such scenario.

While some large e-commerce practitioners such as e-bay, amazon.com, etc. manage the tracking process in-house, others choose to entrust it to specialist tracking service providers, such as AMNs.

If the tracking process is carried out by the e-commerce practitioner in-house, then the available visitor information is limited to the interactions within practitioner's own domain. But as third-party tracking service providers offer services to many e-commerce sites, they can offer additional information for a premium price. Such information could include, e.g., which website did the visitor arrive from, which website did the visitor go to or what products were perused in previous sites, among other useful information. Some service providers offer remarketing leads by using the

information they have gathered in competitor sites that have subscribed to the same tracking service. Using a tracking service provider expands the accessibility scope of visitor data but is still limited to those e-commerce sites that have subscribed to the same tracking service provider.

While some of these external third-party tracking services are known to only track user interaction that relate to visits and transactions and are not offering any additional services based on the data gathered in this process, there are others, who offer additional value-added services, such as re-marketing strategies.

The currently discussed *business analytics* scenario and the *Insights as a service* scenario described in the next sub-section are similar in the sense, that they both are concerned with gathering business insights of the Internet users, enabling informed strategic management decisions in enterprise. But they differ at many levels, that warranted looking at the two scenarios separately. Firstly, the tracking process is carried out by either a single e-commerce company within their own e-commerce site, or by an e-commerce based tracking services provider, who might provide services to a multiple of such e-commerce companies. Secondly, the audience is relatively small, being only customers who visited the e-commerce site of the practitioner or the collection of e-commerce sites that have subscribed to the same tracking service provider. Thirdly, in addition to their browsing behaviour within the specific site(s), the scope of the information gathering is usually limited to a few PII, such as name or contact details. In contrast, in the next section, we look at the practice of large corporates who serve customers globally through their hardware, software or social media products, who have a global reach such as Google, Microsoft, Apple, Facebook, LinkedIn, Twitter etc. They gather large scale of information and create comprehensive “personas” across the globe, with the intention of selling a plethora of services to any future customers they intend to serve.

2.3.3 Insights as a service

OSN service providers, search engines and other large-scale service providers who gather very detailed customer interactions with the Internet belong primary to this category. In 2018, the Cambridge Analytica scandal exposed how OSNs such as Facebook gather large amounts of behavioural data to enrich existing user-profiles. That led them to create comprehensive digital personas to expand their business models from targeted advertising to population influencing business. It further reveals how third-party entities such as Cambridge Analytica were able to further endanger user-privacy and a whole society in general, though stealing data from Facebook to expand the business model to include psychological operations (“psy-ops”) to influence whole societies and populations during electoral and political campaigns (Bakir, 2020; Berghel, 2018; Laterza, 2018; Manokha, 2018; Margaret, 2020; Richterich, 2018; ur Rehman, 2019).

Though *Psy-Ops* business model has been already exposed and many research has been published since 2018, the general population do not appear to grasp the implications; in contrary Afriat et al. (2021) found that many youth consider it is OSN’s right to use the data they gather, to generate revenue. This research will present a privacy model based on privacy intrusion levels associated with different tracking use cases.

2.4 Stateless vs. Stateful tracking

Tracking methods can be divided in to two main categories as stateful tracking and stateless tracking, depending on the underlying tracking technique, which also has an effect on the reliability and accuracy of the tracking process (Englehardt & Narayanan, 2016). All stateful tracking techniques save *state*, in this case a UID in the user’s computer. Stateless tracking techniques do not save any data on a user’s computer, instead uses different techniques, to identify each client-browser uniquely, using data within the HTTP request sent by the client-browser to the webserver.

2.4.1 Stateful tracking

Stateful tracking has a high rate of accuracy, as the webserver saves identifying data such as a UID on the user's computer or device, and every webserver that has access to this user data can accurately and reliably track any user across all the tracked domains. For example, tracking technologies used in AM and other e-commerce applications require a great accuracy in tracking user traffic to reward affiliates with agreed amounts of commissions. Loss of a transaction detail results in the loss of the commission that the affiliate is entitled to, which can lose the confidence in the system. Usually, the identifying data is saved using a HTTP cookie on user's browser. This research explores additional techniques that we discuss in the "Alternative Tracking" sub-section further below, such as local storage provided by HTML5, Flash cookies and ETags, that can be used to improve accuracy and revive expired or lost identifying data (Ayenson et al., 2011). The important characteristics of stateful tracking is, that data is stored on user's device and tracking process is reliable and accurate.

Contents of a cookie can only be read by the domain who owns the cookie. The HTTP cookies that belong to the web site visited by the user, are called "first-party cookies". These are primarily used to improve user experience. A website can also cause a browser to receive HTTP cookies that belong to other service providers of the Internet from other domains. They are primarily used for the purpose of advertising or gathering user demographics for marketing and analytics. These are called "third-party cookie".

2.4.2 Stateless tracking

As the name suggests, this category of tracking methods do not store identifying data on user's device, thereby sacrificing some of the accuracy and reliability of stateful tracking. Stateless tracking uses "browser fingerprinting" as the tracking method, instead of using HTTP cookies (Englehardt & Narayanan, 2016; Laperdrix et al., 2016; Libert, 2015). While there are many browser fingerprinting algorithms (Sanchez-Rola & Santos, 2018), the basic concept is to combine multiple pieces of data provided by the browser in HTTP requests, to generate a single specific identifying data value that can

identify a browser uniquely. When a browser sends a resource request to a web server, it also sends some “header” information, among which it also sends attributes relating to browser capabilities, browser’s and operating system’s configuration data (Laperdrix et al., 2016). On top of this, a browser fingerprinting algorithm can use Asynchronous JavaScript calls to find out and report back further features of the client computer, such as whether Adobe Flash and Java is enabled, what plug-ins are enabled, the list of fonts installed, etc. Algorithm used by Laperdrix et al. (2016) combines 17 such attributes to create their version of browser fingerprint. Though individual piece of data such as the browser version or the OS version is not unique, the combination of those data, together with the IP address is up to 94.2% unique (Eckersley, 2010).

Among other uses, stateless tracking is used in business analytics to generate customer demographics and behavioural information which depend on processing large volumes of tracking data that is often less than 100% accurate in tracking activity. But due to the large volume, the margin of error becomes less significant. As stateless tracking methods gather very large amounts of data, Big Data and AI solutions are often used to process data and gather business insights out of these data. For example, the buying habits of a user, what products were browsed and purchased, or what products were browsed but not purchased, how long a user spent in a specific website or specific pages within one or multiple e-commerce sites enables a marketer to target advertisements and remarketing strategies personalised for that customer (Baumann et al., 2019). Equally, by following an Internet user across multiple sites on the Internet, it allows an interested party to gather large amounts data such as reading interests, political affinity, what topics catches the user’s attention on OSN sites, what posts were liked or shared on Facebook, etc. These data can be combined to create a persona, that allows other processes to predict a lot about the user and even determine what the next user-action might be (Libert, 2015).

Researchers in browser fingerprinting domain would quite rightly claim that the newer techniques used by them allow them to track a user even more reliably than when using cookies. This is true in a

scenario, where a user deletes or blocks cookies, stateful tracking will fail, but stateless tracking can still identify the user with a very high accuracy. On the other hand, when a cookie is present, it allows the stateful tracking process to identify with an absolute accuracy, which stateless tracking does not have. Therefore, within an AM context, it is more important to use stateful tracking, while stateless tracking methods can enhance the reliability during those scenarios when cookie-based tracking fails.

Each tracking technology has its own merits and is suitable for a specific requirement. Business analytics providers and security agencies generally use a mix of stateful and stateless tracking, utilising the advantages of both methods. There are other stateless tracking methods such as *Behaviour-based tracking*, that exploits large-scale host access data, such as queries received by DNS resolvers, which need source and destination IP addresses and the access time, which is outside of the scope of this research (Banse et al., 2012).

This research investigates vulnerabilities associated with HTTP cookie based (stateful) tracking technologies, within two usage domains, namely, within AM environment and within a Business Analytics context. As both domains are important for e-commerce activities and management & marketing purposes of the enterprise, this study examines the impact of the vulnerabilities from the perspective of a commercial enterprise.

2.5 HTTP cookies for tracking

Due to the highest possible accuracy and reliability required in e-commerce tracking scenarios, in this research the scope of study is limited to stateful tracking techniques. Though HTTP cookies were not intended for tracking users on the Internet as discussed earlier, it was designed for *state management*. Tracking a user by multiple domains involves the ability of each domain to access the managed state from a previous interaction, which includes a unique identifier, thus enabling each site to identify a browser uniquely during recurring visits over time. Therefore, any stateful tracking technique is a good potential candidate for online user tracking. Ease of use, i.e., minimum amount of effort needing at

client and server sides to set an identifier and alternatively read an identifier, makes it even more suitable for the purpose. Another criteria that will be evaluated in the experiments will be algorithmic complexity, with least processing time and processing power requirement, that makes a technique more suitable over another. As typically busy e-commerce transaction servers handle large number of transactions per minute, and a tracking server that tracks many such e-commerce server transactions simultaneously need to identify a client-browser at speed, accuracy and with minimum processor load. Simply reading a unique identifier in a cookie or other stateful tracking mechanism is therefore faster and less processor-intensive than the stateless tracking mechanism, that need to build a signature to compare with the collection of such stored signatures on server. Andriamilanto et al. (2021) claim those two hundred and sixteen fingerprinting attributes that are being processed by their algorithm, create on average a dozen kilobytes per signature and takes a few seconds to process. The tracking characteristics of the HTTP cookie is used as baseline in making comparisons with other stateful tracking techniques.

2.5.1 Single-event tracking

In some e-commerce activities, the tracking need is limited to a single event, such as a user visiting a website, an advertisement appearing in user's screen (CPM), a user clicking on an advertisement (click-tracking or CPC), or a user signing up for a membership, email list, a petition, etc. Such single event tracking requirements occur commonly in Internet traffic generation endeavours. An interested party pays a fee for each such event, and the website or the search engine or similar entity that promoted that event to the user, gets rewarded for that event. What the user did after this single event, i.e., which other sites the user visited subsequently, and further interactions between the user and other sites are not part of the traffic generation model, hence no further tracking of the user beyond this single event is needed. In such scenarios, the underlying tracking technology used is simpler, easier to implement, less error-prone and there is a selection of tracking methods to use. For example, our previous research on AM frauds discussed in a previous sub-section looks at different techniques that

can be used for cookie-stuffing fraud. We found a JavaScript can be used to execute a resource request from the click-tracking URL, which successfully records a single event such as a click-even as needed by CPC or an instance of displaying an advertisement in a CMP event etc. But the cookie that is sent to the browser is rejected by the browser, due to cross site scripting (XSS) restriction (Bath, 2011). Without a cookie future visits cannot be traced to the same user. Another user-click on another day will appear as a new visitor, but it is irrelevant for CPC advertising models, as all clicks get remunerated. Nevertheless, cookie-stuffing fraud executed using a different technique, such as using an image file request, causes the cookie to be saved in the client-browser, which enables multi-event tracking.

2.5.2 Multi-event tracking

Tracking multiple events of user interaction need techniques that go beyond single-event tracking. The multiple events can occur within one single browsing session or over many days, weeks or months later. *Conversion-tracking* discussed under the sub-heading *Affiliate Marketing* above is an example of a multi-event tracking scenario. The user first clicks on an advertisement, during which the *click-tracking* event is recorded in a database, and a cookie set in client-browser, but the visitor traffic generation through the click action is not rewarded under the CPA marketing model. Subsequently, during the same browsing session or at a later date, when the user *converts* the visit to a monetary outcome, e.g., by making a purchase, then a *conversion-tracking* instance is generated, then matched with an existing click-tracking record. The affiliate who generated the traffic originally, will be paid a commission. A loss of a single tracking event can result in a monetary loss for the affiliate, as most commission rates are between five to ten percent of purchase price. Therefore, it is vital for such traffic generation methods to track multiple user-interactions accurately and reliably over a predetermined period. The tracking validity period is determined by the lifespan of the cookie as and when it is set. Another example of a multi-event tracking scenario is business analytics services, wherein the set cookies are valid for a much longer timespan, so that during that period, a visit to any

website being monitored by them causes a tracking event to register in the tracking server's database. This allows such business analytics services to profile a person based on the websites visited, frequency and duration spent on specific websites and specific pages, browsing habits etc. Therefore, this research will carry out experiments to ascertain the multi-event tracking capability among the techniques under review.

2.5.3 Single-domain tracking

Tracking within one single domain is carried out by business insights gathering endeavours. This can be carried out by e-commerce sites through their own e-commerce software. Such tracking data can provide information such as the source of the traffic by looking at the *Referer* header, how long a visitor spent on specific pages, what products were perused, what products were purchased in combination, etc. By saving a unique identifier, repeat visits, and buying habits over time can be added to the information-mix. Detailed traffic generation insights cannot be generated if events only within a single domain is tracked. Equally, a third-party cannot be entrusted with the tracking task, with a single-domain tracking scenario.

2.5.4 Cross-domain tracking

Cross-domain tracking (XDT) involves tracking user-interactions across multiple web domains that may be geographically distributed and owned by different entities that do not communicate directly with each other. XDT capabilities are useful for different purposes. Generating network traffic today, happens across multiple websites. A user may click on a product that appear on one website, that causes the visitor to arrive at the e-commerce site that sells the product. In between, the traffic moves through an intermediary site that records and keeps track of the source and destination of the traffic, as the e-commerce site must pay the source for traffic generation. There can be many intermediaries involved in one e-commerce transaction, where each intermediary needs to be rewarded (Baumann et al., 2019; Chachra et al., 2015; Olbrich et al., 2019; Snyder & Kanich, 2015, 2016). Hence this kind of tracking is a technical necessity, as an underlying technology used in different e-commerce activities

(Amarasekara & Mathrani, 2017). Such tracking capability is achieved using “Cookies” or similar methods, that can store a small amount of data to identify a web-user uniquely, which does not capture PII, which therefore is usually not considered to be a privacy threat. The unique identifier is usually a long number or a GUID. The same tracking method can also be used to track web-users for multiple other reasons by commercial and governmental entities. They may capture online behavioural data that is combined with PII to create comprehensive user profiles that invade the privacy of users, without their explicit permission. As both PII and non-PII based tracking use similar technologies to capture data, regulations that restrict usage of such techniques (e.g., using HTTP cookies) can adversely affect scenarios that use tracking only as an underlying technology to manage state. Experiments in this research concentrate on multi-event and multi-domain tracking techniques that are required for e-commerce and e-marketing endeavours.

2.6 Alternative tracking techniques

Stateful tracking and stateless tracking was compared and contrasted above. This research focuses on stateful tracking due to the reliability and accuracy it offers, and due to low resource usage on tracking server, which translates in to low-latency when handling large amounts of transactions. Prior research has presented some alternative techniques to track users in a stateful manner.

As discussed earlier in the chapter, no web standards have specifically been designed for online user-tracking, but state management standards and mechanisms have enabled the use of those state management techniques to track users online. From the inception of the first HTTP cookie RFCs (Kristol & Montulli, 1997) it was known that cookies can be used to track user-activity, and that there were concerns about privacy. Despite repeated attempts to block third-party cookies by default, due to the pressure from advertising industry, which depends on the ability to customise advertisements targeted at users, browsers have so far allowed third-party cookies by default (Kristol, 2001, p. 12). Therefore, HTTP cookies have been considered the de-facto tracking mechanism, and it continues to remain fit for purpose. But other tracking methods that have been discussed in previous literature,

which are being reviewed below and assessed through experiments during this research, have not been originally designed for online user-tracking process. Each technology has been designed to fulfil a different function. Any future developments of those web standards therefore do not guarantee that the usability of it as a tracking technique will be preserved. Hence, utilising alternative techniques will require continued assessment of the new developments of standards and its efficacy and utility as a tracking technique. Next, a review of current knowledge related to the alternative methods that have been used for tracking are presented. In the chapters that follow, the experiments that were designed to evaluate their efficacy as of now, are described.

2.6.1 Flash cookies

Adobe Flash has been used as a multimedia extension in most browsers for a long time before the advent of HTML5. A shared storage named “Local shared objects” (LSO) was provided to store data that is accessible to all Flash content running within different browsers, if a computer has multiple browsers installed, and also accessible to any stand-alone Flash widgets present in a computer (Adobe, 2015). Web applications have been using, since around 2005, adobe’s LSO, under the name “Flash cookies”, to store application specific data and unique identifiers, enabling them to track users in a similar manner as HTTP cookies. Flash cookies have better functionality than HTTP cookies for user-tracking: LSO storage is shared between browsers, and the storage is still accessible even if a user used “in Private” mode. HTTP cookies are not shared between browsers and are not accessible in “in Private” mode. That allows *Flash cookies* to accurately identify a user, even if different browsers installed in a computer is used or uses “in private” mode, while HTTP cookies will identify a user as a distinctly different user in each case, thus failing in the tracking process. Unlike HTTP cookies, the Flash cookies do not have an expiry date, and can store up to 100KB of data compared to 4KB limitation of HTTP cookies. Deleting and blocking techniques used on HTTP cookies had no effect on Flash cookies either (Soltani et al., 2010). Flash cookie is considered to be almost indestructible and has been used to re-spawn deleted HTTP cookies. For additional robustness, multiple tracking methods have been used in

tandem, sharing the same unique identifier across all tracking methods. If a user deletes the identifier, for example if the HTTP cookie with the unique identifier gets deleted by a user, the tracking process can re-create a new HTTP cookie with the identified copied from the Flash cookie, thus respawning the deleted HTTP cookie and making the tracking process robust (Ayenson et al., 2011; Laperdrix et al., 2016; Soltani et al., 2010). Benninger (2006) noted that Flash cookies can be made accessible to multiple domains; they are resistant to clearing of browser caches and the use of it is invisible to the user.

All the above characteristics of Flash cookies would make the perfect candidate for a tracking cookie. But, a newer update of Adobe in 2010 has tightly integrated LSO with browser security settings (Ayenson et al., 2011). Adobe Settings Manager documentation (Adobe, 2015) explains that since the release of Adobe Flash Player 10.1, Adobe supports the “private” browsing mode of web browsers, by disabling access to Adobe’s LSO, thereby acting similar to HTTP cookies under similar circumstances. Similarly, keeping in line with browser behaviour, Adobe clears the LSO data when a user clears browsing history and browser cache. With this update, Flash cookie lost its “super-cookie” status as has been described in earlier research literature.

Since the introduction of HTML5, with support for multimedia, the popularity and the need for Adobe Flash multimedia has further diminished, that by the end of year 2020, the Flash player reached its *End of Life*. Therefore, Flash cookies do not have any relevance and were excluded from our experiments in this research.

2.6.2 Microsoft Silverlight

Silverlight is a Microsoft implementation similar to Macromedia Flash, that included more than mere multimedia capabilities. It was also a rich client application that can run within a browser or as a stand-alone application, which can directly communicate with the server using a multitude of communication protocols beyond HTTP. Microsoft Silverlight client that resided on user’s computer

had access to its own local storage similar to Adobe LSO, which can be used to store a tracking UID that is accessible across browsers (Belloro & Mylonas, 2018; Sanchez-Rola & Santos, 2018). Microsoft announced *End of Life* for Silverlight from 12th October 2021. It no longer supports for Chrome, Firefox or any browser that uses Apple's Mac operating system (Microsoft, 2020). Therefore, Microsoft Silverlight was excluded from our experiments.

2.6.3 HTML5 Local Storage

The "*Local Storage*" introduced with HTML5 is a new client-side state management mechanism, that has many similarities with Adobe's LSO discussed above from a tracking perspective (Hickson, 2021). The purpose of "*Local Storage*" is to provide web applications with the ability to store any user-specific data locally on the user's computer. As such data does not need to be shared with the server, there is not easy mechanism to send the unique visitor identifier back to the server. Therefore, the interaction between the application and the local storage is through JavaScript, using which, the identifier can be extracted from local storage and use one of the few different methods available to send that information to the server.

Ayenson et al. (2011) found in a sample of over 5,600 popular websites that used HTTP cookies, thirty seven sites were using Flash cookies and seven sites were already using HTML5 Local storage for user-tracking, as early as 2011. Belloro and Mylonas (2018) found that HTML5 Local storage was used in nearly 58% of the over 460,000 domains under their study as a tracking vector. Tracking data stored in Local storage does not expire until explicitly deleted, unlike HTTP cookies and even larger storage size of 5MB surpasses the storage size of HTTP cookies and LSO capacity.

2.6.4 ETag

With the introduction of *Entity Tag* (ETag) as a web cache validation mechanism (Fielding & Reschke, 2014), it was discovered that ETags too can be used as a tracking mechanism and therefore are referred to as "*Cache cookies*" in some literature (Ayenson et al., 2011). ETags carry the version

identifier of a specific resource and entrust the client browser to return it with every request for the same resource, back to the server. The server then compares the returned version identifier with the version identifier currently in possession of the server. If they match, the server sends a result code “304 Not Modified”, so that the client browser can use the copy of the resource in its cache. If the ETags do not match, the webserver will send the latest version of the resource. HTTP-cookies and ETags are part of the HTTP protocol, designed for communication of information, hence browsers take over the responsibility of sending ETags back to the server.

Ayenson et al. (2011) found in a sample of 5,600 popular sites that used HTTP cookies, that two sites were already using ETags to respawn blocked or deleted cookies. They note that ETag was still accessible for user-tracking, even if a user was using “in Private” browsing mode, which our experiments found not to be the case anymore.

HTTP cookie and ETags are both designed for communication between the server and the client browser, primarily for the consumption of the server. This is an important distinction that is considered in this research, when evaluating the *ease-of-use* of alternative tracking technologies.

2.7 Privacy concerns

An improved robustness and accuracy in tracking techniques may appear as a more persistent and privacy invasive threat, in the minds of some privacy advocates. Online tracking is fast becoming synonymous with stalking, with increasing number of countries rushing to introduce plethora of new privacy laws. Adhering to multitude of regional and country specific privacy laws on the Internet where physical borders are obscure, and compliance with such regulations is not only difficult, but also is somewhat defeating the purpose of such privacy concerns (Wachter & Mittelstadt, 2019). New research findings suggest General Data Protection Regulation (GDPR) introduced by the European Union in May 2018 (GDPR, 2016) does not achieve its intended purpose, due to click-fatigue (Utz et al., 2019). Papadogiannakis et al. (2021) found more than 75% of tracking activity at websites occur

even before the user was given a choice of how cookies should be used, as per the GDPR requirements. Alternative tracking methods that are not based on use of cookies, which we discuss in this research have been found in extensive use within their large-scale study.

While it is important to protect the privacy of Internet users, it is equally important to develop and maintain robust mechanisms to maintain state in a traditionally stateless ecosystem, across geographically distributed multiple domains, making e-commerce activities reliable. Therefore, it necessitates identifying and categorising different use cases of cross-domain user tracking on the Internet. Such tracking practices span from a purely technological necessity in one end to person-identifying and data-marketing endeavours at the opposite extremity. This segmentation enables practitioners and regulators to define and adhere to regulations and best practices, that would effectively curb privacy intrusions without unintended consequences of technological curtailments. This research examines different technologies that may be used to strengthen the online tracking process, thereby also verifying which of the previously presented technologies are still usable for tracking purpose today, with current developments in technology. Then, it examines different use cases of online tracking and categorises them into levels of privacy intrusion involved and levels of indispensability in terms of a technical necessity. Finally, chapter 4 (Artefact Description) presents how improved and more reliable online tracking techniques can enhance e-commerce activity without compromising privacy of Internet users when used purely as an underlying technology. Chapter 6 (Discussion) also reveals which techniques have what levels of intrusions, when combined with PII. This knowledge will provide clarity to policy developers and legislature to formulate effective and consistent regulations and policies without undermining the technical necessities of legitimate e-commerce activities. It will also facilitate practitioners to define boundaries in their implementations. Importantly, the scientific community can extend this research to develop technological solutions and frameworks that can automate machine-to machine negotiation processes, protocols and standards

between client and server while adhering to privacy guidelines, thus eliminating human intervention that leads to “click-fatigue” (Utz et al., 2019).

Most web traffic generation methods involve a minimum of three web domains. For example, organic or paid searches (e.g., with Google) would involve the Google domain, an e-commerce domain, and the visitor domain. Apart from online traffic generation endeavours, business analytics and customer demographic data services also require XDT capability (O’Brien et al., 2018). Usually e-marketing services gather behavioural data on customers, such as origin of the traffic, total vs. successful visit counts, products perused by customer, time duration spent on different pages and other customer demographic information that helps marketers to target marketing campaigns to specific audiences. They also provide helpful insights for a marketer to understand if the customer needs are met by their product offerings.

If the tracking process is carried out by the e-commerce practitioner in-house, then the available visitor information is limited to the interactions within practitioner’s own domain. But as third-party tracking service providers offer services to many e-commerce sites, they can offer additional information for a premium price. Such information could include, e.g., which website did the visitor arrive from, which website did the visitor go to or what products were perused in previous sites, among other useful information. Some service providers offer remarketing leads by using the information they have gathered in competitor sites that have subscribed to the same tracking service. Using a tracking service provider expands the accessibility scope of visitor data but is still limited to those e-commerce sites that have subscribed to the same tracking service provider.

The hierarchical nature of the information access capability of various service providers enables information exploitation to occur at different degrees. As one traverses up the hierarchical tree, service providers sitting at a higher level have increasingly wider visibility. Services at the top of the hierarchy have visibility over the largest number of node sites. Almost every Internet user utilises some

form of a service provided by at least one of the largest global service providers such as Google, Facebook, Microsoft, Apple, or similar tech giants. Often a person may be using services of all or most of the above tech giants. Being on top of the hierarchical tree, they have visibility of user-interaction over most of the Internet (Schelter & Kunegis, 2016). To use services provided by these tech giants, one needs to create a user profile with personally identifiable information and sign-in with a user account. A cookie that is placed into the site-visitor's web browser during the sign-in process will identify the visitor uniquely across all services offered by these tech giants and at numerous other seemingly independent websites. Often the presence of these tech giants is not directly visible to the visitors of a third-party website. But, unbeknown to the visitor, most third-party websites utilise some services of these tech giants in the background, such as resources from a Content Delivery Network (CDN), widgets or subscription to a business analytics service. When such a resource is loaded to the browser while rendering the third-party web page, the cookie set by the tech giant is automatically sent back to the web server with each new request. That reveals the presence of the user at the specific third-party site, thus allowing such services to gather data on user's navigation across the Internet. When using a browser application provided by one of these tech giants, the exposure of the user data increases even further, as the browser can monitor all interactions with websites, without depending on the cookies. Using operating systems or hardware (e.g., phones, tablets) provided by these tech-giant has the highest exposure, as the personally identifiable information are available at the operating system level (Narayanan & Reisman, 2017). Previous research found that 80 percent of Alexa's top one million websites were being tracked by Google, while another found the percentage to be even higher at 97 percent, among the top hundred websites (Ayenson et al., 2011; Libert, 2015). Starov and Nikiforakis (2018) found news and sports related websites, followed by shopping and recreation related sites were more commonly tracking and fingerprinting users than adult sites, children's sites and those belonging to "Computer" category.

Business Analytic services such as Google Analytics (Universal Analytics) offer standard services free of cost to everybody, while charging a price for premium services. The comprehensiveness of the insights sold as premium services depends on their ability to track users across the entire Internet (Krishnamurthy & Wills, 2009; Narayanan & Reisman, 2017; O'Brien et al., 2018; Schelter & Kunegis, 2016). Therefore, many such service providers offer free services with limited features to users who are not willing to pay for those services. This in turn will allow a provider to harvest comprehensive set of user related data of a large customer base, that makes up the product which will be marketed as a premium service.

Some of the free services that are offered by such operators are: web browsers, e-mail services, cloud storage, business analytics, widgets such as counters, exchange rate and weather information, CDN services, DNS services and others. The information exploitation mantra is simple: place as many cookies on the client browsers as possible by offering shared resources through CDNs or provide as many free services as possible, since it will enable the service provider to place a cookie and gather as many “pings” along the way.

Integration of third-party Software Development Kits (SDK) are considered a best practice in software engineering discipline, to implement commonly and often used security related functionality, such as online payment systems, cryptography, analytics among many others, as rigorous, well-tested, modular and reusable libraries. Feal et al. (2020) argue that the use of such SDKs in Mobile applications comes at a privacy cost for end-users as they do in web and desktop applications. This happens as current mobile operating systems allow these third-party contents to run within the same context and with same privileges. Though the users may have authorized the privileges to host app, such third-party contents that are invisible to the user do not thereby inherit those permissions.

Another research stream follows development of comprehensive online anonymity, while still preserving certain customisation. Mor et al. (2015) argue that privacy and personalisation are not

mutually exclusive. *Bloom cookies* proposed provide the ability to create privacy-preserving compact user profile managed by the client to limit the exposure of client privacy, while still providing sufficient information for a search engine to personalise search results.

Roesner et al. (2012) proposed a framework to classify third-party trackers to five groups based on how they manipulate the browser state, observable by client-side behaviour. In chapter 6 (Discussion), this research presents a privacy model based on information seeking behaviour of the tracking actors and resulting privacy intrusion levels, which can be used by technical and policy initiatives.

Chapter 3. Methodology

Many previous research studies exposed the use of alternative tracking vectors by using different research methods, from a client perspective. Some used crawlers to access the most popular websites and empirically observed how web servers thus visited placed UIDs in various client-side local storages (Ayenson et al., 2011; Buhov et al., 2018). Others analysed large sets of browsing data or web search logs (Mor et al., 2015) and browser plug-ins such as “SoThink, FoxTracks, WTPatrol” (Mittal, 2010; Soltani et al., 2010; Yang & Yue, 2020) to detect patterns that lead to such discovery. This provided great insights into the prevalence of new tracking vectors and the scale at which they are used from a client-side perspective, but they do not provide details of technical implementations on the server side. Basically, they show us what is being done, but now how it is being done by the server. This research is aimed at experimenting the use of these tracking vectors within a multi-domain and multi-event tracking environment such as in an Affiliate Marketing Network. It needed a real-world environment with access to server-side software implementations, where unrestricted access can be gained to carry out the experiments.

The Internet-simulating network environment AMNSTE, which was used for the experiments during my previous master’s degree research work provided an ideal solution, that needed to be further customised for this research. The hardware configuration of the network and the software implementations are discussed in detail in subsection 3.2. This solution offers the complete flexibility of unhindered ability to control the server-side implementations that provided the client-side perspective that was discussed above, which was used by other researchers.

The first goal of this research is to evaluate the currency of the alternative tracking vectors discussed in previous research literature. Tracking techniques that are known to be obsolete among practitioners in the industry are still been discussed in some information science research, hence this

research expects to update status quo with regard to tracking vectors (Wang, 2018). Further, all the three research goals presented in section 1.2 require hands-on live experiments to be carried out on the Internet or within a similar multi-domain network. Hence, simulation of real-world scenarios within a lab-environment and evaluation through empirical data was the method adopted to solve the research problem.

3.1 Selecting a research paradigm

Information science research are characterised by two research paradigms: behavioural science and design science (Hevner et al., 2004; March & Smith, 1995). While behavioural science research usually contributes to theory, design science research (DSR) are primarily concerned with creating new artefacts to solve existing problems and challenges in the industry (Hevner et al., 2004). The challenge is not knowledge *transfer*, but knowledge *production* (Holmström et al., 2009). Goldkuhl (2004) argues that techniques used in behavioural sciences can be used in design science, and Holmström et al. (2009) find that both research strategies can be used in tandem. They explain that most conventional research are explanatory in nature and seek to study phenomenon that already exists. In contrast, design science research is exploratory in nature, where a given problem-space is investigated to develop a solution, thereby creating artifacts as artificial phenomena, that will then be evaluated and studied. Holmström et al. (2009) argues that exploration and explanation are not mutually exclusive, instead highly complementary. Exploration research produce artifacts that can be studied by explanatory research.

An industry problem that required improved reliability of the underlying tracking systems in e-marketing technologies, due to technical limitations and due to fraudulent activities, gave rise to this research project. DSR paradigm appeared as the best choice, being a pragmatic research paradigm, focused on creation of innovative artefacts to solve real-world problems. DSR is highly relevant to information system research because its focus is creating design artefacts combined with relevance in

the application domain. The design ideas are then communicated as knowledge to relevant information system stakeholders and communities (Hevner & Chatterjee, 2015).

As field studies enable behavioural science researchers to understand organisational phenomena in context, the process of constructing and exercising innovative IT artifacts enable design science researchers to understand the problem addressed by the artifact and the feasibility of their approach to its solution (Nunamaker et al., 1991). March and Smith (1995) identify two design activities and four design artifacts produced by DSR. The two activities, build and evaluate are iteratively carried out until the evaluation process satisfies the solution is fit for purpose. The artifacts thus created are constructs, models, methods, and instantiations. The outcome of a DSR can be any of the four artefacts or combinations thereof. *Constructs* are described as the descriptive language of the problem- and solution-space, which are presented in chapter 1 and 2.

Models depict the problem and solution space, allowing us to understand the connection between the two. The privacy model presented in chapter 6 represents different levels of privacy intrusions caused by web applications based on their privacy-information seeking behaviour, as privacy related problem-space. The analysis and description of it enables development of the solution-space through implementation of software artefacts and regulations that govern their implementation.

Methods are algorithms, the steps carried out to reach the solutions. They can be defined algorithms or an informal description of an approach of a solution or both. Chapter 4 presents the *Method artefacts* created as outputs of this research, that are presented as sequence diagrams, which are accompanied by a description of the processes. Based on the experiments that were carried out on AMNSTE2 test-environment, they present how the selected alternative tracking techniques can be used as tracking vectors individually and in combination to form a more robust tracking technology.

Instantiations are prototypes, proof of concept or similar working systems, enabling concrete assessment of efficacy and utility. At the conclusion of this research, the Method artefacts were

instantiated as functional prototypes, that are publicly accessible to researchers and practitioners on the Internet. The URLs to access each *Instantiation* is provided with *Method* descriptions in chapter 4.

3.2 Choosing the test environment

The design of AMNSTE, which was used by this researcher during a previous research project was chosen, as experiments in this research require a similar multi-domain network that need some further extensions and newer software artefacts (Amarasekara & Mathrani, 2015). The new extended test environment was named AMNSTE2. The hardware implementation and the network topology of AMNSTE and AMNSTE2 are same. They both simulate the Internet, with same technologies in use, appearing identical to the application layer. But server applications were only designed to use HTTP cookies as tracking vectors. Therefore, server-side applications for each of the different participating domains were extended for them to be able to use alternative tracking vectors within experiments.

The exploration of the research problems required:

- a) testing the validity of non-cookie based tracking capabilities mentioned in previous research literature and selecting those that are still current and functional
- b) Testing the efficacy of any valid candidates for multi-even cross domain tracking
- c) Testing the privacy intrusion behaviours and testing efficacy as a privacy-preserving tracking vector
- d) Attempt to discover unique identifiers within HTTP request object that can be used as a tracking identifier

Usually, most web-related experiments can be carried out within an integrated development environment (IDE) such as Visual Studio, which enables hosting multiple software projects within one solution. When debugging the software solution, each project is hosted in a separate webserver instance at a different port of the *localhost*, simulating multiple applications or even multiple domains. But experiments in this research need to be tested within an environment subjected to cross domain

restrictions such as Cross-Site-Scripting (XSS) and Cross Origin Resource Sharing (CORS) restrictions, as in real-world. Localhost with different port numbers does not simulate multiple domains, hence the requirement to create physically separate domains was recognized.

A network topology of an AM Network was chosen as such network can fulfil the requirements of all network topologies that are needed for different experiments mentioned above. Experiments relating to different web-traffic generation models in e-commerce, business insights gathering, experiments relating to third-party business analytic service providers and privacy related experiments all of which can be carried out in such a multi-domain network. An AM network requires minimum of four different domains. The topology and technical details are described later in this section.

Complete and unrestricted access to the tracking infrastructure of all domains of an AM network was required to execute the experiments to answer our research questions of how to make the HTTP cookie based cross-domain tracking system more robust and also to what extent the business data is exposed during business insights gathering process. As a result, Affiliate Marketing Network Simulation and Testing Environment (AMNSTE2) was developed using the same technologies as a real-world AM Network, which is described in detail below.

An experiment on cross-domain tracking (XDT) requires multiple domain-based networks on separate IP segments that are interconnected with same network technologies and topologies to simulate Internet infrastructure. To track visitor-interactions across multiple domains, all the domains being tracked require the ability to communicate with a mutually available central tracking domain. From the XDT scenarios discussed above, a simulation of an AMP was chosen for this experiment, which comprises a minimum of four separate domains. Such network allows us to test different XDT based technology implementations. The setup can simulate different e-marketing models such as display advertising, CPC model or revenue-sharing models such as CPA. It can also be used to simulate business analytic services, CDN's and other multi-domain transactions.

AMNSTE2 is not a single web application. It is a collection of four different categories of network domains. While they share a standard domain-based network configuration, each category is unique based on the bespoke web applications installed in each category of servers. Each application was developed as part of this research and each abstracted to the minimum requirements for the specific category. Multiple instances of some categories of domains were needed for the experiment, e.g., multiple instances of affiliates were needed to experiment a real-world scenario of an AM network, where a tracking server can accurately identify which affiliate generated the traffic among multiple affiliates. Multiple instances of e-commerce servers were required to experiment the capability of the tracking server to identify transaction at the specific e-commerce domain in a real-world scenario, where once tracking service provides services to multiple e-commerce domains. It was further necessary for experiments based on business analytics gathering and for experiments on privacy issues. It enables us to verify the techniques that allows a tracking domain to gather interactions of one user across multiple domains across the globe. When creating new domains for our experiments, we could use any arbitrary domain names without any consideration for the existence of such domains in real-world, as our experiments are carried out in total isolation away from the Internet. Yet, the choices of the domain names were first checked on the Internet which allowed me to port the network setup on to the Internet, without changes, by acquiring those domain names on public Internet.

The findings could be generalised for non-secure HTTP environment and for secure HTTPS environment by carrying out our tests in both networking environments. It necessitated me to develop different server software that implemented different tracking techniques for secure protocols, as the techniques for non-secure protocols were not usable in an HTTPS environment. research.

3.3 Hardware configurations

My choice was to create the test network using virtual servers due to ease of setting up the network and adding and removing new servers and complete domains at will, without additional hardware costs and time-consuming configuration requirements. Considering that most server environments on

the Internet are hosted on virtualised hardware, it was expected that the technological implications of conducting experiments on a virtual network would be similar to a hardware-based network environment. but initially, yet a physical hardware-based network based on the same network topology was set up to replicate the same set of experiments. Though multiple server applications can be developed and tested as multiple projects within a single solution in development environments such as Microsoft Visual Studio IDE, they would not simulate cross-domain restrictions, accurately. This prompted me to build a hardware-based network environment, parallel to the virtual hardware-based network environment, enabling me to carry out experiments in both environments, at the initial stages, until it was evident that both environments return same results.

3.3.1 Hardware-based test environment

As shown in Figure 6 four desktop computers with Intel Core I5 processors and 8 GB Random Access Memory (RAM), 500GB Hard disk drives (HDD), with Microsoft Server 2016 *Datacentre Edition* operating system were used. Each was configured as a Primary Domain Controller in separate IP segments. A CISCO Catalyst 2950 switch with VLANs were used to separate the IP subnets and a CISCO 1841 Router in a “Router on a Stick” configuration was used for inter-VLAN routing; i.e., to route traffic between the subnets.

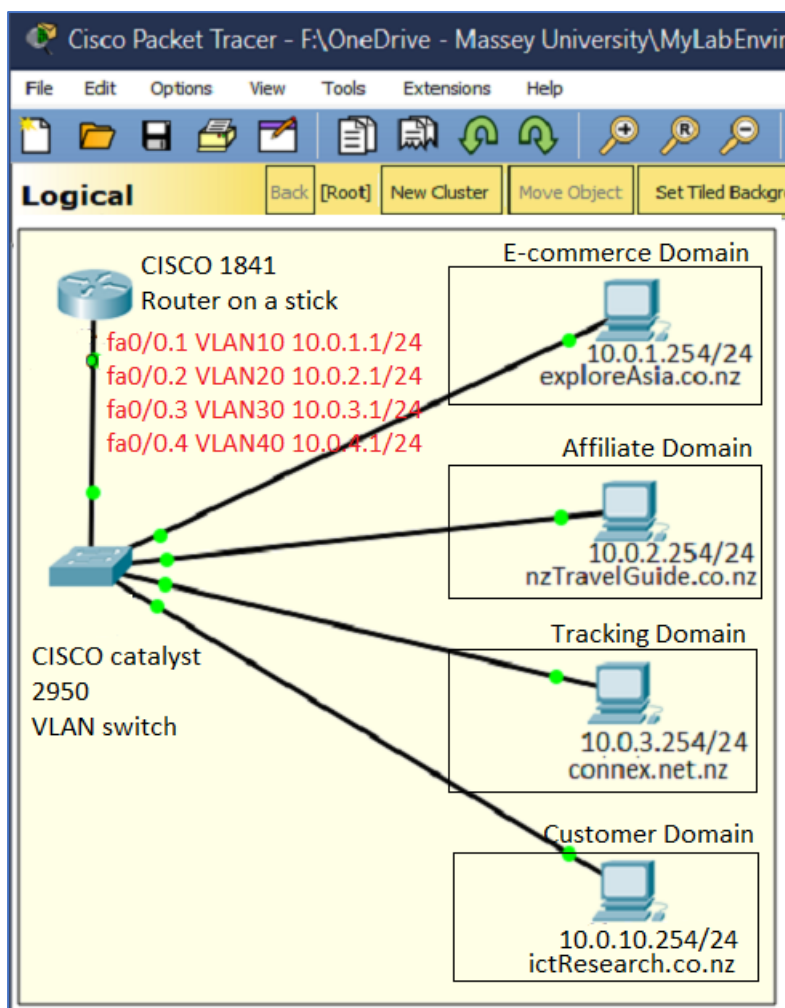


Figure 6: AMNSTE2 hardware-based network topology

3.3.2 Virtual networking infrastructure-based test environment

Virtualized infrastructures were created on my physical laptop computer, that had an Intel Core i7 processor, 64GB of RAM and 1.5TB of Disk space using Solid State Disk (SSD) Drives running Windows 10 Enterprise edition. Microsoft Hyper-V was used to create virtualized servers; with parent virtual machine (VM) running on Microsoft Server 2016 Enterprise edition operating system. Microsoft SQL server 2017 was added, with Internet Information server (IIS) enabled for webhosting, and Domain Name Server (DNS) for name resolution. This parent VM was cloned to create all other servers used in the experiments. After cloning, as shown in Figure 7, the newly cloned server was raised as a Primary Domain Controller (PDC) within a new Domain, IP configuration was set to a new subnet within the 10.0.0.0/16 subnet, essentially making each subnet only accessible through routed traffic, confirming to the real-world topology of the Internet. Each domain was connected a dedicated virtual switch,

created with Microsoft Hyper-V. A new clone of the parent VM with routing services enabled, was used as the network router.

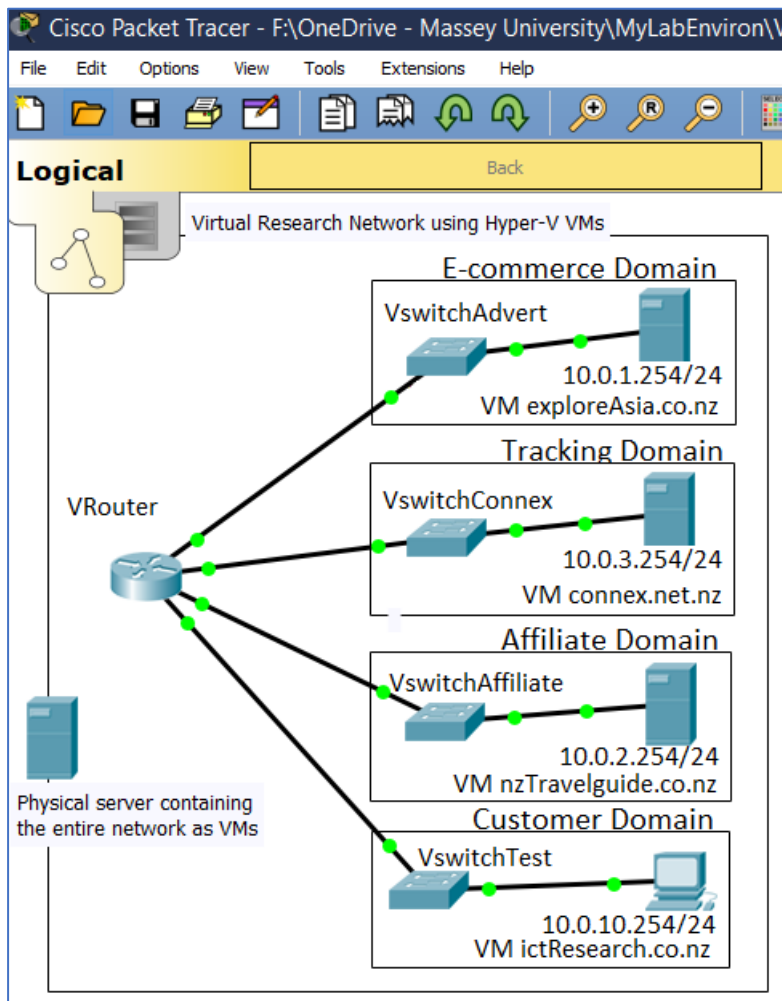


Figure 7: AMNSTE2 network topology with virtual infrastructure

3.3.3 Internet-based public test environment (Public-AMNSTE)

At the end of the experiments, it was not practical to make the complete network infrastructure, which makes up AMNSTE2, available to researchers. We configured multiple publicly accessible webservers on Internet to take different roles such as tracking servers, affiliate websites and e-commerce sites, using the software that we developed for those specific roles. I will refer to this publicly accessible network as “Public-AMNSTE” within this thesis, to avoid confusion with fully fledged internal network environment AMNSTE2.

For different experiments, we need different number of servers from each of the three categories. Public web URLs for some of the servers are listed in Table 6. Most tests are carried out at affiliate websites. In an AM strategy, that is where the web traffic is generated, for the e-commerce servers. The home page of each affiliate website describes what AM is and offers links to different categories of experiments.

Table 6: Domain name and categories of Public-AMNSTE hosted on Internet

Affiliate domains	e-commerce domains	Tracking domain	Protocol
https://newzealandtravel.net.nz https://nztravelguide.org.nz	https://exploreasia.co.nz https://ecotourismpng.com https://bestcars.ecopng.com	https://connex.net.nz https://cnx.ictresearch.co.nz	Secure HTTPS
http://unsec.nztravelguide.org.nz	http://ecovillagerundu.com	http://technicalfrontiers.co.nz	HTTP

After ascertaining which tracking techniques discussed in previous research literature were still relevant and which of them were useful for cross-domain multi-event tracking, we have demonstrated how each tracking technology can be used singularly as well as in tandem with other tracking vectors, creating a “robust tracking” system. Same domain names that were used in the simulation environment was used on the Internet.

3.4 AMNSTE2 (System design and Implementation)

A typical AM network has four categories of independent domain networks, each category with a distinct role or function. In a real-world environment, each function will be fulfilled by a full-fledged application with many application features and aesthetics that go beyond minimalistic tracking requirements. For example, an e-commerce application will have a fully-fledged shopping cart of some sort, with many features, customer profile with purchase histories, etc. Each application is abstracted to minimum, but with all essential functionality to allow examination of technical aspects of the

simulation. The minimalistic page contents enable us to concentrate on core technology areas without distraction. Each application implements the same real-world tracking technology and tracking processes and a back-end transactional database records all the transactions, which allows us to inspect the outcomes of our tests. The tracking process is observed by the researcher by activating “developer tools” view on the browser by pressing “F12” key and examining the HTTP request and response headers. A detailed description of the AMNSTE2 prototype and how tracking process works, has been presented in a previous paper (Amarasekara & Mathrani, 2016).

The AMNSTE2 represents a visitor traffic generation model for an e-commerce site, on the Internet. The four categories of actors are described below, in the sequence they occur during a traffic generation action:

3.4.1 Internet user / Researcher Domain

This category represents Internet Users who are referred to as visitors or customers within an AM context. This category is represented by the *Customer Domain* (Figure 7). Within our experiments the researcher represents the visitor, and we describe privacy issues as it relates to such visitor. Unlike other three categories, this category is not represented by a specific web application. We consider visitor as a separate web domain, as the visitor is connected to the Internet through an Internet Service Provider (ISP), therefore the IP address, domain name and geographic location of user appears to the public as a network node of the ISP. The IP address of the visitor’s device will be assigned statically or dynamically through ISP’s DHCP server. Different Server and Desktop VMs running on different operating systems and different browsers were connected to the 10.0.10.0/24 subnet, to act as an Internet user for different experiments. The only requirement is the availability of one or more browsers on the device; multiple browsers can confirm that the results are compatible between current browsers.

3.4.2 Affiliate website

This category is represented by one or more independent third-party affiliate websites, who have undertaken to motivate its site-visitors to visit one or more e-commerce sites. Visitors are usually enticed with special offers, for products or services that the e-commerce sites sell (Figure 8). This is usually done by displaying a banner advertisement to click. The Internet traffic generation process starts with a visitor loading the homepage of the affiliate website on to the browser and clicking on a banner advertisement.

Most experiments take place on the affiliate sites, as we study different tracking techniques and privacy vulnerabilities. Each home page represents a typical website of an affiliate, which are usually static HTML pages with text and multimedia content that interest followers of such pages. The homepage of affiliate website shown in Figure 8 has three different banner advertisements and hyperlinks between text descriptions, each representing three different e-commerce sites. By clicking on the banner advertisements, a researcher can explore in real-time, how the tracking system works. Each click in affiliate's domain takes the researcher to the e-commerce site that is represented by the advertisement. Though it appears to the user that only the two websites, i.e., affiliate and e-commerce site were involved, the results page located at tracking service provider domain shows the click record, which happens in the background transparent to the user.

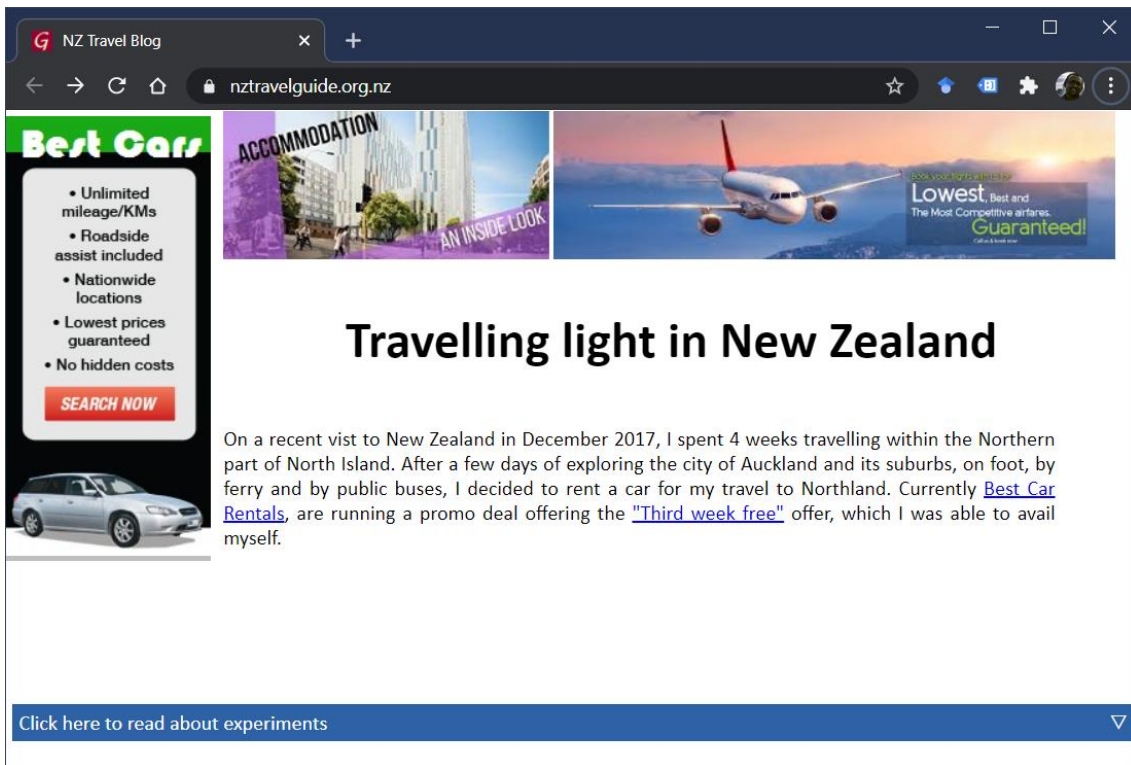


Figure 8: Affiliate home page

Some home pages were created using pure HTML, to demonstrate that most of the fraud can be executed using simple HTML pages. A few home pages were created using active server-side page implementations, which enabled me to add additional capabilities dynamically, such as changing the user-agent field with each web request to avoid detection by the tracking server (Chachra, 2015).

The collapsed bottom half of the affiliate homepage as shown in Figure 9 explains how to carry out the experiments. The two web links *Technologies* and *Frauds* give access to a dedicated page each for the two groups of experiments.

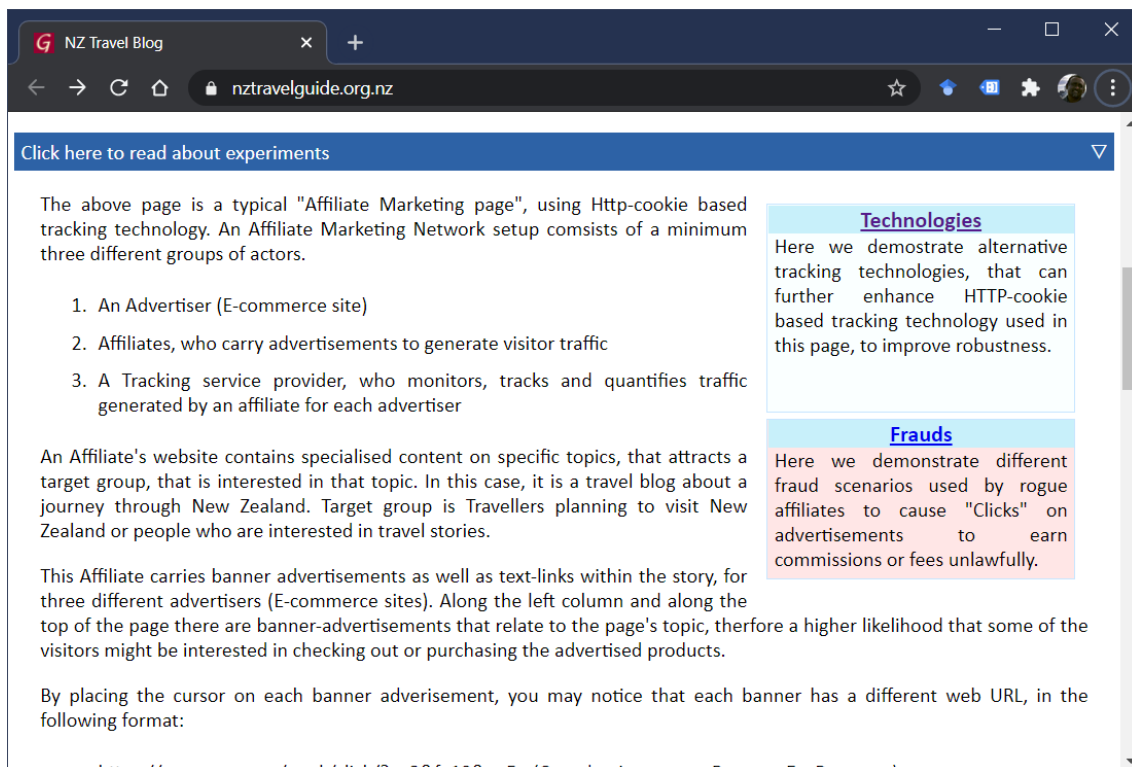


Figure 9: Links and descriptions of experiments

The *Technologies* page shown in Figure 10 was created to demonstrate alternative tracking technologies that we experimented with successfully. In turn, each of the three links, i.e., *Local storage*, *ETags* and *Robust Technologies* opens a page each, that describes how the technology works, and how to carry out the experiment to check the efficacy of that tracking method, which we describe in detail in the Chapter 4, under Artifact Description. *Local storage* page demonstrate how the Local storage provided by HTML5, can be used instead of HTTP cookies for tracking. *ETags* page demonstrate the use of cache control mechanism ETags discussed in subsection 2.7.3 above, as the tracking mechanism. The Robust Technologies page demonstrates our recommendation of using a combination of HTTP cookies, HTML5 Local Storage and ETags used in tandem as a robust mechanism that can depend on one technology, when another fails.

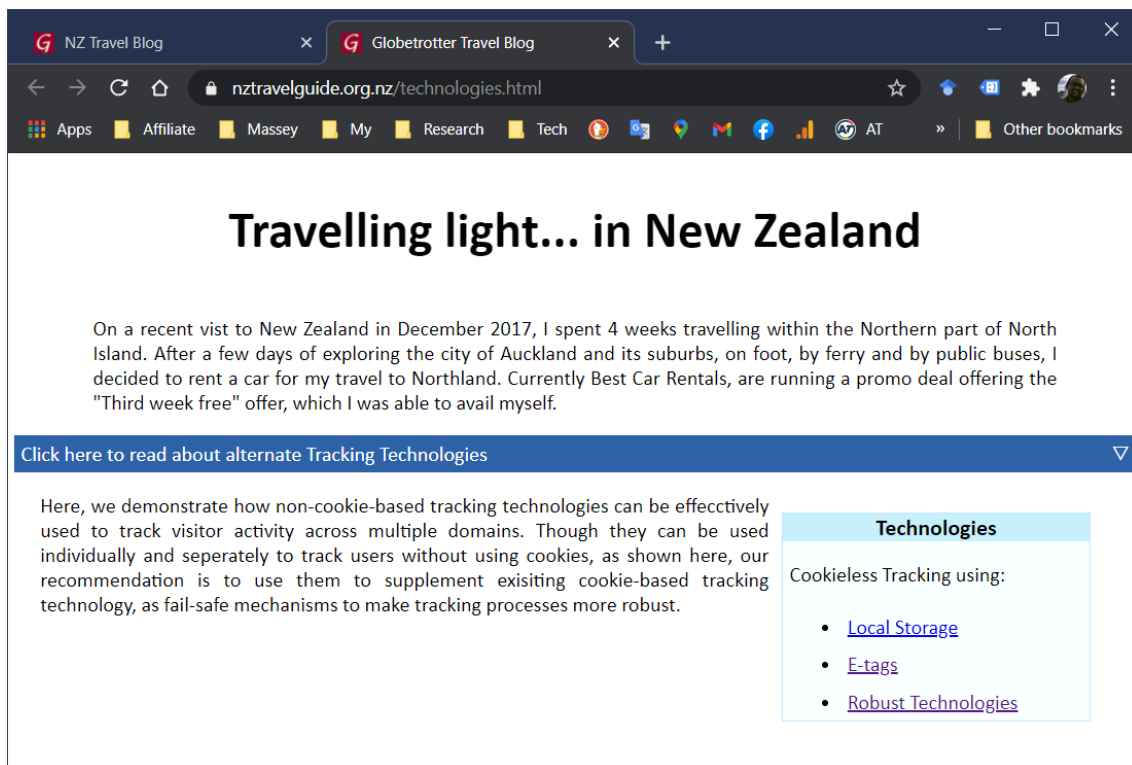


Figure 10: Tracking Technologies Group

While each page describes the tracking process with the specific technique, the researcher can click a banner advertisement and verify the click-tracking result in real-time by observing the tracking record in the tracking results page, shown in Figure 11, under *Click-Results*. The banner click takes the researcher to the e-commerce site, at which, if the researcher makes a purchase action, a payment confirmation page will be presented with the receipt ID. After which the researcher can verify the conversion tracking result in the *Conversion Results* table in Figure 11. A successful conversion result will show payment amount, and most of all the Tracking ID, that correspond to the *Click-Tracking ID* in Click-Results table of the same page. The process can also be followed using the *Developer Tools* provided by the browser, by invoking F12 in most browsers.

ICT Research Group Home Results About Contact

Click-Results

Tracking ID	Click Date	CookieID	Affiliate ID	Advertiser ID	Offer ID
11577	1/27/2021 5:22:05 PM	637474117257332439	20	3	4
11576	1/27/2021 3:50:43 PM	637474061765741819	20	1	22
11575	1/27/2021 3:50:01 PM		20	1	5
11574	1/27/2021 3:49:58 PM		20	1	5
11573	1/27/2021 3:49:56 PM		20	1	5
11572	1/27/2021 3:49:53 PM	637474061765741819	20	1	25
11571	1/27/2021 3:49:49 PM	637474061765741819	10	2	10
11570	1/27/2021 3:49:48 PM	637474061765741819	10	2	15
11569	1/27/2021 3:49:47 PM	637474061765741819	10	3	4
11568	1/27/2021 3:49:46 PM	637474061765741819	10	1	1

Conversion Results

Conversion ID	Conversion Date	CookieID	Tracking ID	Transaction ID	Advertiser ID	Affiliate ID	Offer ID	Am
144	10/29/2020 9:42:18 AM	637396079565629110	11238	255	1	20	5	10.0
143	10/29/2020 12:40:04 AM	637395753032805096	11237	254	1	20	5	100

Figure 11: Tracking results

The banner advertisement and the hyperlinks in the text descriptions point to the tracking network AMP at <http://connex.net.nz/click.ashx> while affiliate, offer and advertiser Identifiers are passed as parameters of the above URL. A click on the banner or hypertext link will register a click at AMP and places an identifying cookie on the browser and forwards the user to the e-commerce site. The click tracking process is invisible to the user as it happens very fast, usually within milliseconds, which gives the impression to the user that the click caused the browser to take the user to the e-commerce site instantly. The researcher can observe the complete process by activating developer mode on the browser and examining the HTTP request and response headers.

3.4.3 Tracking service provider

An AM network usually has one tracking service provider, with multiple e-commerce sites subscribing to their tracking services and many affiliates being involved in generating Internet traffic.

The single tracking domain was assigned the domain name Connex.net.nz, which is at the centre of all the tracking activities in this study. The tracking server contained a bespoke software that had the function and ability to track user activities within all other e-commerce domains. “Pixel-codes” embedded in the webpages belonging to e-commerce and e-marketing sites cause visitor-browsers to “ping” the tracking server at connex.net.nz. This enabled us to test tracking service capabilities for AMNs based on different AM models, e.g., display advertising, click advertising and revenue-share advertising models. Different service endpoints were created to offer different services which are discussed later in this section.

Apart from the server configuration mentioned above, a tracking application was developed as part of this research, that can track user activity at other domains. The application has different service endpoints to track click-actions at affiliate sites and conversion-actions at e-commerce sites. Further, different tracking techniques such as those using HTTP cookies, Local Storage, ETags, and the robust tracking (combination of different techniques) were assigned dedicated endpoints. A detailed description of those tracking techniques is presented in the *Artifact Description* section below.

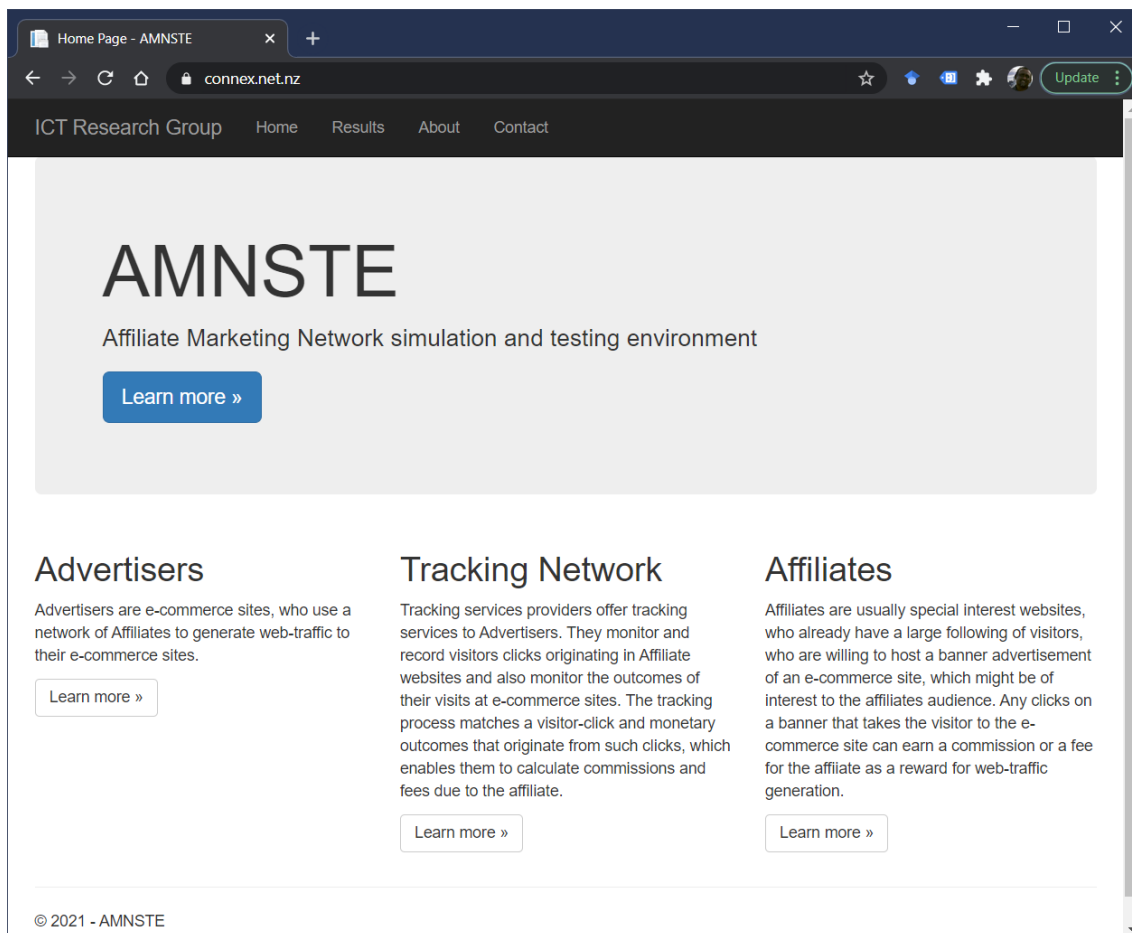


Figure 12: Tracking Server Homepage

The experiments do not usually need to access the tracking server directly as it is only used by tracking processes for machine-to-machine communication, through web services. Nevertheless, a simple home page was created for the publicly accessible site, providing links to the starting points of the experiments, viz. to affiliate websites. A real-world tracking application usually provides access to a portal where their clients (e-commerce sites and affiliates) can view and interact with historical data on tracking activity as well as access reporting facilities. Though all cross-domain tracking related data is saved to a database in this tracking domain, we have provided access to the tracking data, at the affiliate websites, as the researchers carry out most experiments there. In addition, this application also provides access to the same tracking results similar to Figure 11, through the *Results* menu option as shown in Figure 12. The researchers can verify the results of each experiment by ascertaining the *click* and *conversion* data on this page. A more comprehensive data analysis and inspection can be

carried out by connecting to the underlying Microsoft SQL database directly. The tracking service providing AMP is located at: <http://connex.net.nz>

3.4.4 E-commerce sites

E-commerce servers have in addition to the standard domain configurations, an e-commerce application that was specifically developed for this research experiments. Usual functionality of an e-commerce application was abstracted to a minimum set of functionalities needed for the tracking experiments.

One or more independent e-commerce site, each within a separate Internet domain, who wish to subscribe to a network of *affiliates* who can generate visitor traffic to their e-commerce sites, for an agreed fee or commission. E-commerce sites are used in experiments pertaining to finding alternative tracking mechanisms in our quest to make the tracking process more robust and resilient. In business intelligence gathering experiments, these domains represent Small to Medium Enterprises (SMEs) that would either implement functionality within their e-commerce software for insights gathering or those who subscribe to services such as Google's Universal Analytics. In privacy intrusion detection experiments, this category represents SME's who might gather information in a privacy pervasive manner, or expose privacy intruding data to third-party services, inadvertently.

Home pages of e-commerce sites simulate a simple products page displaying the products it sells. Our experiments need to capture the tracking process that takes place at the time of payment and payment confirmation stages. Hence, we have not added an elaborate shopping cart function, instead the researcher will manually enter the total price in the text box and press the "Pay" button, simulating a payment action. The transaction is recorded in the transaction database of the e-commerce site and a payment confirmation page is generated with a "conversion *Pixel*", a piece of JavaScript code. While presenting a confirmation page, with the total price paid and invoice number to the customer, the *conversion-Pixel* triggers a conversion-tracking action in the background, which will cause the tracking server to receive the transaction data.

The researcher clicks on an item as many times to add as many items to the shopping cart; or types an amount in the total field and presses the “Pay Now” button to emulate a payment action. That causes the application to record the transaction in the sales database of the e-commerce site, which can be viewed at the “/sales.aspx” page. The application then returns a payment confirmation message at the bottom of the page, with a hidden “Conversion Pixel” which causes AMP at <http://connex.net.nz/conv.ashx> to register the “conversion” of the visit to a sale. The tracking process here, consists of passing advertiser identifier, the total price and transaction identifier to the tracking AMP transparently to the user as there are no visual clues to the user, about the background process.

3.5 Simulating privacy intrusions

Improved robustness in tracking technologies has a causal relationship with privacy intrusions of Internet users, in the eyes of users, privacy campaigners and privacy regulators. This research aims at demonstrating how the robustness of the tracking process can be improved, while preserving the privacy of the user, thus demonstrating that tracking and privacy are not mutually exclusive, contrary to the popular belief.

The two main groups of experiments in this research are “robustness improvement-based” and “privacy based” experiments. All *robustness improvement-based* software artefacts developed and described under 4.1 subsection demonstrate improved reliability while preserving the privacy of the user. They are implemented without using PII, but a simple UID for tracking.

The purpose of the “privacy-based” set of experiments is to demonstrate how the privacy of an Internet user can get compromised, and to what degree by different use cases. In this research, use cases are broadly divided in to five categories, based on their information seeking behaviour and thereby the resulting privacy intrusiveness in their real-world implementations. The amount and the richness of the privacy data that can be gathered within each category differ, based on the technological limitations within which each category of applications operate. Privacy-related

experiments represent these different categories of operations and demonstrate how much personal data can be gathered at each implementation level they operate. In chapter 6 (Discussion) I present a tracking privacy model that describe the hierarchical nature of tracking use cases that have a positive correlation to the level of privacy intrusion.

Privacy intrusion related experiments can also be simulated using the AMNSTE2 network topology, with the addition of a few simple purpose-built software artefacts. Though click-tracking process can be used for privacy related tracking experiments, same algorithm was applied to a new URL endpoint, to keep the two processes separate. The visitor tracking URL on publicly available version of our test environment is at <https://cnx.ictresearch.co.nz>.

3.5.1 Test case scenarios

Following are a list of privacy related experiments that demonstrate tracking use cases based on their privacy intrusiveness in ascending order.

AM model

This use case represents the least privacy intrusive category when the intention of the application is simply to provide cross-domain tracking capability as an underlying technological necessity but does not capture any privacy related user information. This group of use cases include AM models, e-marketing and other web traffic generation models, which promote user traffic in third-party domains and lead them to e-commerce sites. The experiments under section 3.3 demonstrate this category, hence no additional test setups were needed. The data generated through robustness improvement-based experiments demonstrate non-privacy invasive tracking in one end of the invasiveness spectrum. In the next section, the results of those tests will be analysed to demonstrate privacy preserving nature of this tracking use case.

Local Business insights gathering

This use case represents a scenario, where an e-commerce or any other website gathers business insights based on the visitor interactions with this single website, without association of any external third-party services. A separate URL endpoint for visitor-tracking is implemented, thus separating the e-commerce software from tracking software artefacts. A “*Pixel*” is embedded on all the pages that need to be tracked. A JavaScript *Pixel* or an HTML *Pixel* or both can be used for this purpose. An HTML *Pixel* is HTML code that invokes a resource request from the tracking URL. Originally it was an image element of the size of a single *Pixel*, which is not visible due to extreme small size, that was placed on the HTML page. As described previously, it can be either an image element, or any other resource request, such as a CSS or JavaScript file, iframe, multimedia element or any such element that invokes a resource request from the tracking URL endpoint.

Third-party Business Analytics offering

This scenario is similar to the above, except the e-commerce site wants to gather a richer set of business insights, therefore uses a third-party business analytic (BA) service provider such as Google’s Universal Analytics service. A similar tracking *Pixel* as in previous experiment, provided by the BA service provider is embedded in each page that needs to be tracked, with the URL of the *Pixel* source set to the tracking URL of the BA service provider. Through experiment, we examine the level of visitor information breach and unintentional data leakage of the e-commerce site within a network of one e-commerce domain connected to the BA service. Then we extend the network to include multiple e-commerce domains connected to one BA service provider to conduct further experiments to examine further data spillages to enterprises and privacy breaches to the visitors of those enterprise websites. Such network topology represents a real-world scenario, where many e-commerce sites subscribe to a single BA service provider.

Above different network topologies and configurations enabled me to simulate real-world scenarios in a Lab environment and validate my findings through simulations. The Software artefacts that

demonstrate the techniques which enabled the described outcomes, are next described in the following chapter.

Chapter 4. Artefact Description

Design Science research outputs fall into four types of artefacts, viz. *constructs, models, methods and instantiations* (March & Smith, 1995). *Methods* are defined as algorithms or informal description of an approach of a solution, or both. Multiple iterative cycles of *Design* and *Evaluation* have produced the artefacts that are described in this chapter. They represent alternative tracking techniques that were successfully implemented as tracking vectors during the experiments. The techniques can be used individually as tracking vectors, but they produce more reliability and robustness when used in combination with HTTP cookies.

These *Method* artefacts are presented as Sequence diagrams, accompanied by an informal description. This research has also produced a set of *Instantiation* artefacts, by producing functional prototypes of alternative tracking solutions, implementing method artefacts presented in this chapter, and making them publicly accessible on the Internet. These instantiations enable researchers and practitioners to observe the described tracking techniques and their behaviour to verify validity through test results. *Method* artefact descriptions are accompanied by the URLs for the *Instantiation* artefacts in the following subsection. All experiments in this research were carried out on AMNSTE2, the purpose-built multi-domain network environment, which consists of a multitude of servers, switches, and routers in a virtual setting, emulating the Internet. Since it is not practical to make the complete network infrastructure available to researchers, multiple publicly accessible webservers on Internet were configured to take different roles such as tracking servers, affiliate websites and e-commerce sites, using the software that was developed for those specific roles. In this chapter, the *Method* artefacts are accompanied by the public URLs of their *instantiation* artefacts, where available.

Some artefacts described in the section 4.1 below are alternative tracking methods that can be used for single and multiple-event tracking. They can improve the robustness of the tracking process, which aligns with the research goal 2. The algorithms presented here are privacy preserving tracking techniques, which do not gather PII of the user. They add to the existing knowledge base as theoretical contributions.

4.1 Artefacts relating to alternative tracking methods

Each subsection below describes a tracking method as an artefact, resulting from this research study. They share a few fundamental characteristics of the HTTP protocol, that leads to a set of common behaviour that need to be implemented using different techniques. During a browsing session, the website that was visited by the user is the “first-party”; for clarification, the user represents the “second-party”, but seldom used as a terminology. If the webpage has any third-party resources embedded, the user’s browser will send additional HTTP resource requests to those “third-party” webservers requesting those resources. In return, the user may receive third-party cookies, together with the requested resource from the third-party, which will be accepted and stored by the browser, by default. Usually, the tracking process is carried out by a domain, that is neither visited by nor visible to the user, therefore the tracking domain too is a “third-party”. The HTTP protocol defines some security-related “same-origin” restrictions when accessing content from a *third-party*, viz. Cross-site scripting (XSS), Cross-Origin resource sharing (CORS), etc. (Bath, 2011). Most of the tracking processes take place across different web domains; therefore, are usually subject to “same-origin” restrictions. This research presents algorithms that can track user-activity across the domain boundaries.

During a cross-domain tracking process, the first-party website embeds a suitable type of *Pixel*, provided by the third-party tracking site, within its webpage. A *Pixel* is a customised JavaScript or HTML code block, that causes the browser to make a HTTP resource request to the tracking URL, enabling the tracking process. The placement of the *Pixel* depends on the tracking need. For privacy related experiments they were placed to trigger during the page-load event, which tracks everybody

who arrived at that website and at those specific pages where the tracking *Pixel* was placed. If the need is simply to track a visitor at a website, the tracking *Pixel* can be placed on the home page. By adding the *Pixel* to every webpage basic business insight can be gathered about visitor interactions with the website (e.g., what pages were visited, in which order, how long the visitor stayed in each page might indicate what content is more appealing to visitors, etc.). For CPA model of AM, the *Pixel* was placed to trigger when a user clicks a banner advertisement. When the *Pixel* is triggered, the browser will send an HTTP resource request to the third-party tracking server, transparent to the user. The tracking server thus creates a new tracking entry, using the information that was sent with the HTTP request including the UID. In case of multi-domain tracking, each participating domain, needs to embed a similar resource *Pixel* from the tracking server. A multitude of resources that can be embedded in a webpage to trigger the tracking process (e.g., a multimedia resource such as image, video, sound, or JavaScript, CSS file, or most commonly an iframe) have been demonstrated at <https://nztravelguide.org.nz/test.html>,

Tracking techniques that make it possible for a server to recognize a client-browser within a single browsing session at minimum, are suitable for single-event tracking scenarios. For example, display advertising (CPM) and CPC models only need to track one single event. But multi-event tracking techniques such as CPA advertising model or business insight gathering scenarios are more complex and require the capability to track a client-browser reliably, on every visit over a longer period. The HTTP cookie-based tracking method makes such unique identification easy. The tracking server sets an HTTP cookie with a UID on the client-browser during the first visit. The client-browser will always send the cookie back to the server, with each subsequent connection to the server, which is a part of the implementation of HTTP protocol (Fielding & Reschke, 2014). If the HTTP request is not accompanied by a cookie, it is usually safe to assume that it is a first-time visitor to the tracking site. The following subsections provide detailed descriptions of each tracking technique that can be replicated by researchers and industry partners or integrate into their software solutions.

4.1.1 HTTP cookie-based tracking

The HTTP cookie-based tracking system is used as a baseline for the experiments of alternative tracking vectors in this research. HTTP cookies are part of the HTTP protocol, therefore are the standard method for maintaining state, where a UID is part of that managed state (Kristol & Montulli, 1997). Cookie-based tracking has been used for a few decades. Single-event based tracking scenarios such as CPC and CPM, which only need to track a click action or a banner display event during the page-load event on the browser, are easy to implement. Most of the tracking techniques can be successfully used for single-event tracking scenarios. But CPA which has gained popularity more recently, needs a multi-event tracking model, to track banner advertisement-clicks and subsequent monetary conversions, that need to be reconciled with a corresponding click-record. Therefore, multi-event CPA tracking capability was considered as the baseline for our alternative tracking experiments. The HTTP cookie-based tracking method is presented first, as certain details such as placement of the *Pixel* on a page, and how it functions, alongside other processes are common to all algorithms. The sequence diagram for HTTP cookie-based tracking technique is presented in Figure 13, and processes involved described in Table 7.

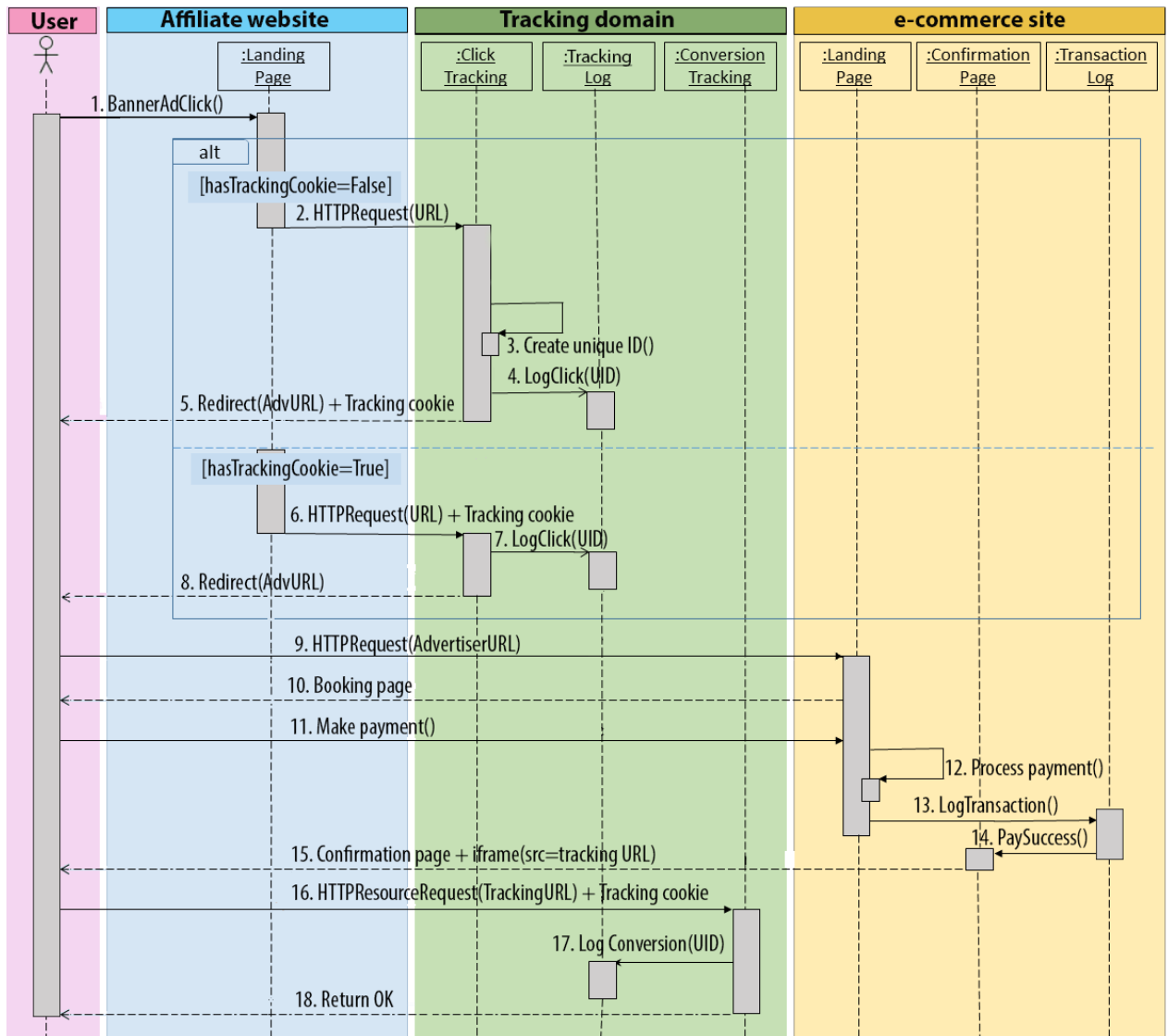


Figure 13: Sequence diagram for HTTP cookie-based tracking process

Table 7: List of HTTP cookie-based tracking processes

Process No.	Process description
1	User clicks on the banner advertisement. If the browser already has a tracking cookie, from a previous tracking event, the program flow jumps to process 6.
2	The click on the banner sends an HTTP Request to the click tracking URL of the tracking domain.
3	Tracking server extracts the affiliate ID, and advertiser ID from URL parameters and creates a new UID for the user.
4	A new click-tracking entry is added to the clicks table with IDs created/extracted in the previous step.
5	Server returns a 302 status code redirecting to the URL represented by the advertiser ID. The response is accompanied by an HTTP cookie with the UID stored. A tracking services provider may provide tracking services to many advertisers and many affiliates, and one affiliate may have multiple banner advertisements from different advertisers. Hence, advertiser ID is used to determine the redirect URL. Program execution continues to process 9.
6	Continuing from process 1, the click on the banner sends an HTTP Request to the click tracking URL of the tracking domain, along with the tracking cookie from previous visit, which was found in the browser's cookie cache.
7	Tracking server extracts the affiliate ID, and advertiser ID from URL parameters and UID from the accompanying cookie. An asynchronous process adds a new click-tracking entry to the tracking database.
8	Server returns a 302 status code redirecting to the URL represented by the advertiser ID. This ends the conditional program flow.
9	The browser sends an HTTP request to the Advertiser's (e-commerce site) landing page.
10	The user may browse through the product catalogue and add products to the shopping cart, which are not shown in this diagram, as they do not relate to the tracking process, to minimize clutter. Finally, the invoice is produced to the client as shown with Booking Page action.
11	The user clicks Pay button.
12	The payment is processed
13	The sale is logged in to the transaction database of the e-commerce site
14	A confirmation page is generated, with a hidden iframe embedded that will be used for conversion tracking process that occur next. The iframe's source property is set to the tracking server's Conversion tracking URL. Advertiser's ID, total payment, and transaction ID of the payment, are appended to the URL as parameters.
15	The payment confirmation page with the hidden iframe, is returned to the browser.
16	Loading the payment confirmation page in the browser triggers the conversion tracking process. It sends an HTTP request to the iframe's URL, which is located on tracking server, requesting iframe's content. The purpose of the iframe is to cause the browser to connect to the conversion tracking URL, with the sales data, enabling the tracking server to track the conversion.
17	The tracking server extracts sales data; viz.: Total paid, transaction ID, Advertiser ID, which was sent by the e-commerce server as URL parameters. The UID of the current user is extracted from the accompanying cookie. All above data is combined to create a conversion tracking entry in the database. In this process, the tracking

	server looks up for the click-record, with a matching UID of a user and an advertiser ID. If found, the Affiliate represented by the Affiliate ID of the click record will be rewarded for the sale, either with a fixed fee or with a commission based on the total payment of the sale. If no click-record matching UID and Advertiser ID is found, it denotes an organic sale; a sale that did not originate through AM model, but a direct sale.
18	The tracking server returns a 200 OK status code, to complete the communication.

4.1.2 HTML5 Local storage-based tracking

This subsection demonstrates the use of Local Storage provided with HTML5 as a cross-domain multi-event tracking technique, without the use of cookies or any other tracking techniques in combination.

The sequence diagram (Figure 14) demonstrates the processes involved.

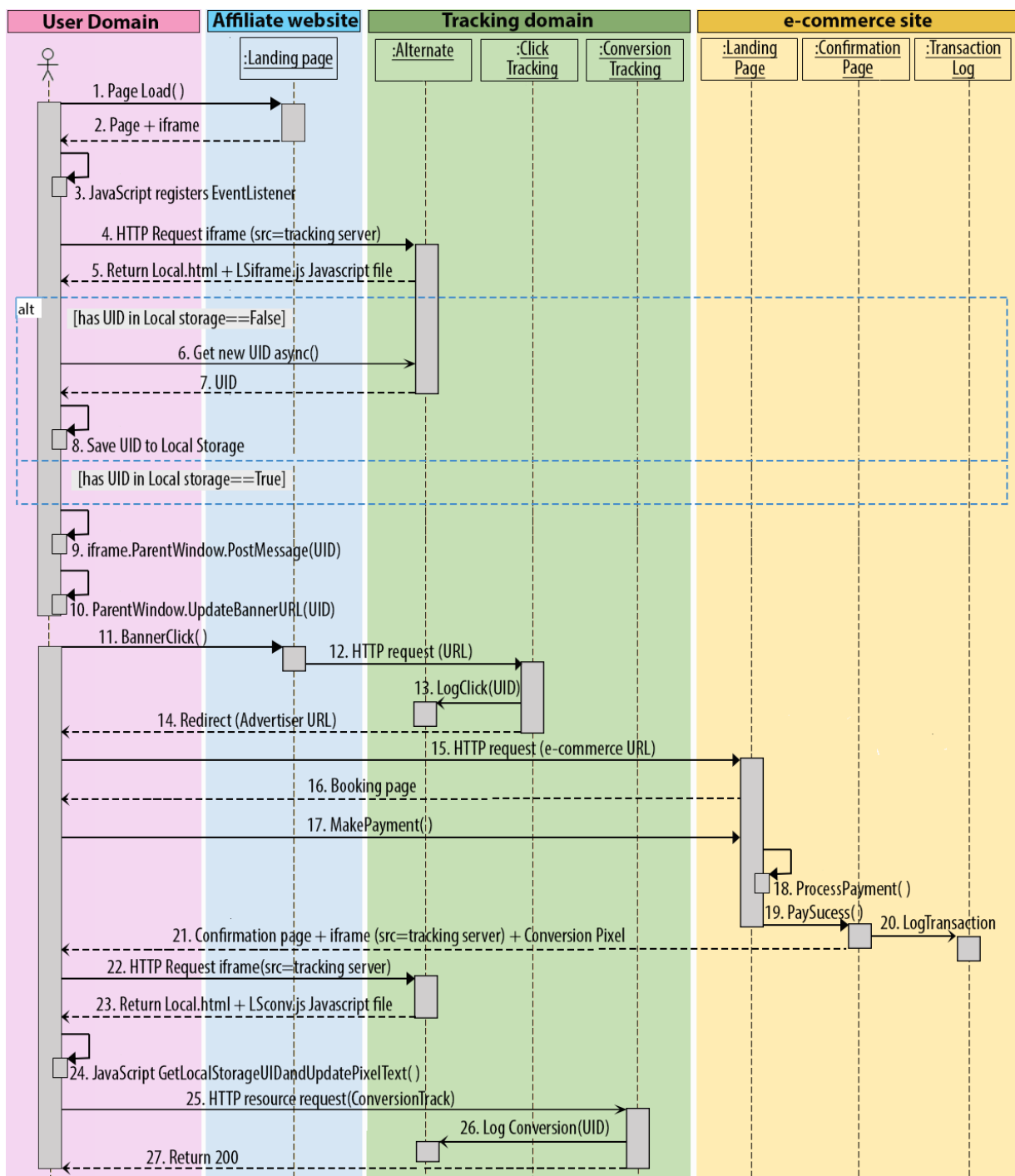


Figure 14: Sequence diagram for Local Storage based tracking process

With this tracking method, the UID will be saved in the local storage instead of a HTTP cookie. When using HTTP cookies for tracking, the UID is always available to the webserver with each HTTP request, simply by reading the UID from the accompanying cookie. But Local storage is a client-side technology, therefore webserver do not have direct access to the local storage of a browser. Hence, whenever a tracking server needs to find the identity of a client during a web request, it needs to first send a JavaScript file embedded in the requested HTML page to the client; the JavaScript will run during the page load event on client browser and read the UID from the local storage and sent it back to the tracking server. Table 8 describes the processes involved in HTML5 Local storage tracking technique.

Table 8: List of Local storage-based tracking processes

Process No.	Process description
1	The user requests the affiliate’s landing page by typing the URL in to the browser, clicking a link or using a saved bookmark.
2	The client-browser receives the requested webpage that has the affiliate page content and an iframe, which is usually set to hidden, as its sole purpose is to track the user without any visible content. The parent page and the iframe both have JavaScripts attached to them.
3	The JavaScript in parent page registers an EventListener to listen to the messages sent by the iframe.
4	The browser continues loading remaining resources in parent HTML page, by next sending for the source of the iframe. The source of the iframe is not located at the affiliate domain, but an URL endpoint in the tracking domain. A cross-site request is sent to the tracking server.
5	The tracking domain returns an HTML page with a JavaScript embedded in it. The hidden IFRAME loads the HTML, but it remains invisible to the user. Next the embedded JavaScript runs within its source IFRAME and looks for an existing UID within local storage. If a UID is found, program flow jumps to process 9.
6	As shown in the Alt interaction frame, if no UID is found in the Local Storage, current user is considered to be a new user. Another asynchronous request for a new UID is made to the tracking server.
7	A new UID is created on tracking server, saved to database, and returned to the browser.
8	The UID is saved in the local storage.
9	As this UID checking process takes place when the landing page of the affiliate’s website is loaded to the client-browser, every new visitor to the affiliate site receives a UID, that is saved in Local Storage. Next, the JavaScript in IFRAME notifies the parent page of the UID for the current user.
10	The URL of the banner received initially at page-load, contained the click tracking URL and Affiliate ID, and other optional parameters such as campaign ID, banner type ID etc., except UID. The Parent page appends the

	UID to the URL of the Banner (e.g., https://connex.net.nz/clicktrack/?advertiser=1&affiliate=3&UID=918273645). This completes the page loading process. User browses the page.
11	User decides to click on the banner advertisement.
12	A click on the banner sends an HTTP Request to the click tracking URL of the tracking domain.
13	Tracking server extracts the affiliate ID, UID and advertiser ID from URL parameters and logs an entry in the click tracking table.
14	Server returns a 302 status code redirecting to the URL represented by the advertiser ID. A tracking services provider may provide tracking services to many advertisers and many affiliates, and one affiliate may have multiple banner advertisements from different advertisers. Hence, advertiser ID is used to determine the redirect URL.
15	Next, the browser sends an HTTP request to the Advertiser's (e-commerce site) landing page.
16	The user may browse through the product catalogue and add products to the shopping cart, which are not shown in this diagram, as they do not relate to the tracking process, to minimize clutter. Finally, the invoice is produced to the client as shown with Booking Page action.
17	The user clicks Pay button.
18	The payment is processed.
19	A confirmation page is generated, with a hidden iframe embedded that will be used for conversion tracking process that occur next. The iframe's source property is set to the tracking server's Conversion Pixel generating URL, together with Advertiser's ID, payment total and transaction ID of the payment as parameters. The Advertiser ID, payment total and transaction ID are part of the Conversion Pixel, which is the set of information that is sent to the Conversion URL, to track the conversion.
20	The sale is logged in to the transaction database of the e-commerce site
21	The confirmation page with the hidden iframe and a partially formed Conversion Pixel is returned to the browser.
22	To complete the conversion tracking record, the Affiliate ID and UID of the user, both of which are unavailable to the e-commerce site, are still required. Due to "Same-Origin" restrictions, to extract the UID from local storage, a JavaScript received from tracking domain is required. Hence, the hidden iframe requests its source document from the tracking server.
23	The tracking server returns the source document with another JavaScript from tracking server, that would enable the script to extract the previously saved UID from Local storage.
24	The internal process initiated by the JavaScript on browser, reads the UID from Local storage and appends and completes the partially formed Conversion Pixel.
25	The JavaScript within the <i>iframe</i> then triggers the Conversion Pixel causing it to send an asynchronous HTTP request to the conversion tracking URL.
26	The conversion tracking process that takes place within the tracking server, is the same for all the different tracking scenarios discussed in this section. The conversion tracking process looks up the Click Tracking table, to find the click record with matching UID and Advertiser ID. If found, the Affiliate

	represented by the Affiliate ID of the click record will be rewarded for the sale, either with a fixed fee or with a commission based on the total payment of the sale. If no click-record matches the UID and Advertiser ID, it is considered an organic sale; a sale that did not originate through AM model, but a direct sale. The conversion record is saved to the database.
27	The tracking server returns a 200 OK status code, to complete the communication.

4.1.3 ETag-based tracking

Though ETags were introduced as a cache management mechanism, as part of the HTTP specification, they can be used as a reliable tracking vector that remains accessible when other tracking vectors fail.

As a tracking vector, their implementation defers from specification recommendations, which is demonstrated below (Figure 15) and processes described in Table 9.

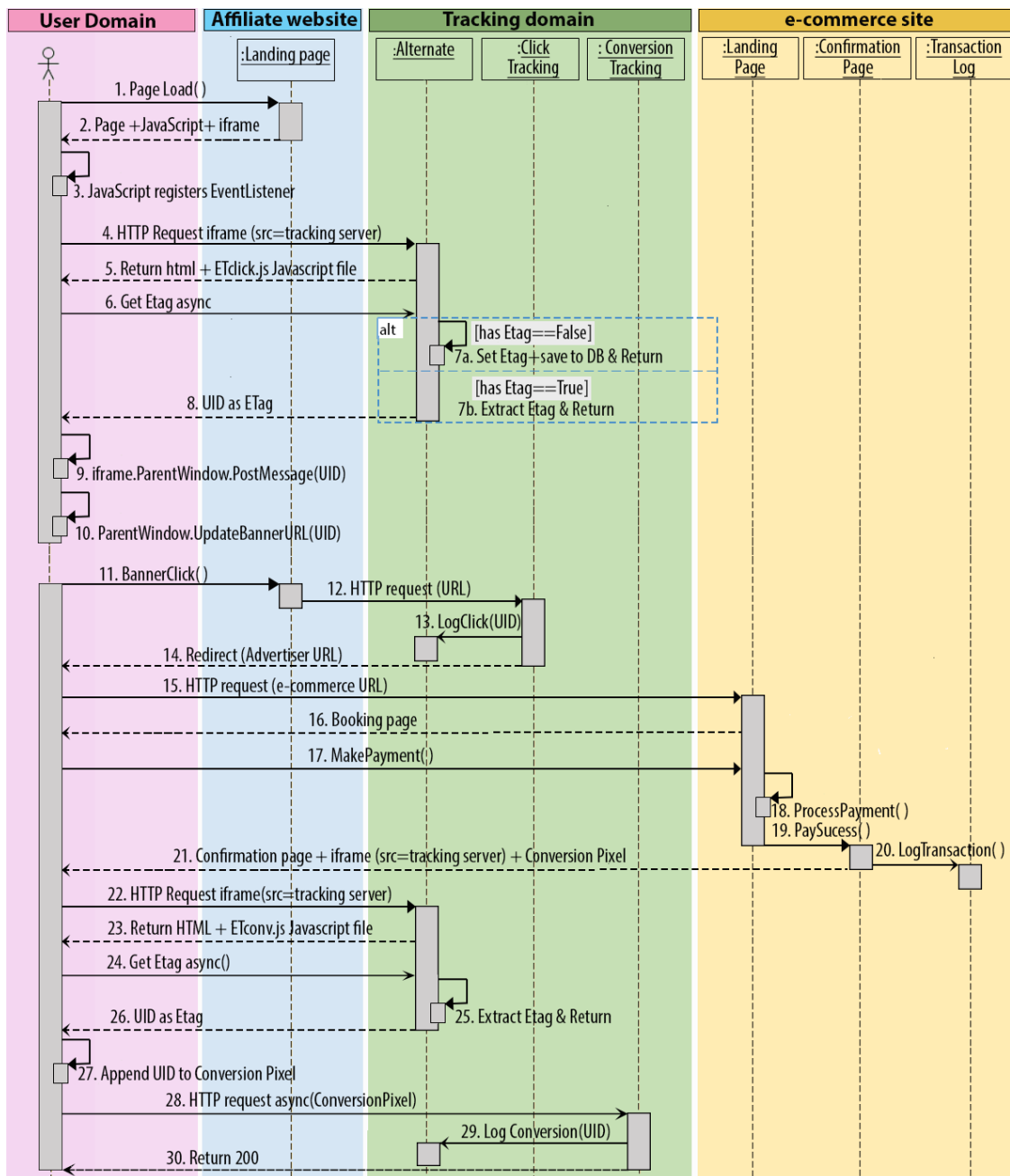


Figure 15: Sequence diagram for ETag based tracking process

Table 9: List of ETag-based tracking processes

1	The user requests the affiliate’s landing page by typing the URL in to the browser, clicking a link or using a saved bookmark.
2	The client-browser receives the requested webpage, that contains an invisible iframe. The parent page and the iframe both have JavaScripts attached to them.
3	The JavaScript in parent page registers an EventListener to listen to the messages sent by the iframe.
4	The browser continues loading remaining resources in parent HTML page, by next sending for the source of the iframe. The source HTML page of the iframe is not located at the affiliate domain, but at the tracking domain. A cross-site request is sent to the tracking server.
5	The browser receives HTML source for the iframe, which contains a JavaScript file that causes the next process.
6	The JavaScript within the iframe causes the browser to send an asynchronous HTTP request to the tracking server
7	If the request header does not contain an ETag, it denotes a first-time visitor. Create a new UID, and save the UID in database and set the UID as the ETag for this resource URL, so that any future visits will return the set ETag, thereby enabling the tracking server to recognize the visitor. If an ETag header is present, send the ETag in a hidden form field, back to browser. This process is necessary, as the browser has no reliable way to read the ETag stored in the browser, associated with the current URL. To read the ETag, the browser needs to send it to the server, and the server can read the ETag, if present.
8	Tracking server returns the UID back to the browser in a from field or as text accessible to client-side JavaScript.
9	The JavaScript code within the iframe that received the UID posts it to the main HTML page within the parent window, as a message
10	The main page that contains the banner advertisement, appends the received UID to the click URL
11	The User clicks the banner, which causes the browser to send an HTTP request to the Tracking server’s click-tracking URL, which is the source of the banner URL.
12	The click-tracking process, extracts the UID, Affiliate ID and Advertiser ID from the parameters list of the HTTP request URL
13	An asynchronous process registers a new click-action in the database, against the user.
14	The tracking server returns a 302-redirect response code with the URL denoted by the advertiser ID
15	The browser sends a new HTTP request to the advertiser’s URL returned by the tracking server
16	The landing page of the e-commerce site (advertiser) is returned to the browser
17	Usual user interactions, such as browsing the products and adding them to the shopping cart are not shown as they are not relevant to the tracking process and to minimize the clutter. Instead, the user enters a total price and clicks “pay now” button, emulating a payment action
18	The payment is processed and confirmed.

19	Payment process on e-commerce server generates a confirmation page, with an iframe. The iframe has an attached JavaScript and within its URL, transaction data required to create a “conversion-Pixel” as parameters. They include total price, receipt number and advertiser ID.
20	An asynchronous process at the e-commerce server records a new transaction within its database, without blocking the main conversion main process of generating tracking <i>Pixel</i> and the confirmation.
21	E-commerce server returns a confirmation page, showing payment data, and a conversion <i>Pixel</i> data within the URL of a hidden iframe.
22	Browser requests the HTML page, which is the source of the iframe from tracking server
23	Tracking server returns the HTML page with an attached JavaScript and with embedded tracking <i>Pixel</i> composed of the data that was passed on withing the URL of the iframe. The tracking <i>Pixel</i> contains conversion tracking data available to the e-commerce server, but still lacks the UID of the user at this stage, which will be fulfilled in process 27.
24	The JavaScript with the iframe makes an asynchronous call to the same URL that generated the ETag in step 6.
25	The tracking server extracts the UID that formed the ETag from the HTTP request headers and stores in a from field or between two <div> tags, making the ETag accessible to the JavaScript on browser.
26	The UID that was stored in the ETag header is returned to the browser
27	The JavaScript within iframe receives the UID and appends it to the URL of the tracking <i>Pixel</i> code. At the time of generating the tracking <i>Pixel</i> the e-commerce server had access to data fields such as Total cost, Receipt ID, Advertiser ID, but not the UID of the user. Hence, this process adds the UID to the URL, making all the information needed for the conversion tracking process, available
28	A further asynchronous call is made to the conversion tracking URL, with the above data
29	The conversion tracking process queries the click tracking table of the database for a matching click event, using UID, and Advertiser ID. If a matching click record is found, it provides the affiliate ID that promoted the click within its website. Based on the total price, the commission amount for the Affiliate Is calculated and conversion tracking record is saved to the database. If no matching click-tracking record is found, it denotes a direct sale, organic search, or another traffic generation model, not AM model.
30	The tracking server returns a result code 200 and thus terminates the click and conversion tracking process successfully.

4.1.4 Robust tracking

This tracking process combines multiple stateful tracking methods that were tested through simulation and presented above on their own merit. By combining HTTP cookies, HTML5 Local storage and ETags as the Robust Tracking method, this subsection demonstrates that more reliable and robust capabilities can be achieved. Should one method fail due to a technical glitch, the tracking process can fall back on another method. Should a user intentionally disable one method, there is a higher chance a different method still stays active, as each of the three tracking methods have their own vulnerabilities that are mutually exclusive.

The Robust Tracking has higher number of processes, as three different tracking methods are being used in combination. Therefore, the processes that take place during page-load event and processes that involve click and conversion actions have been shown in two different sequence diagrams (Figure 16 and Figure 17). The processes involved are described in Table 10 and Table 11.

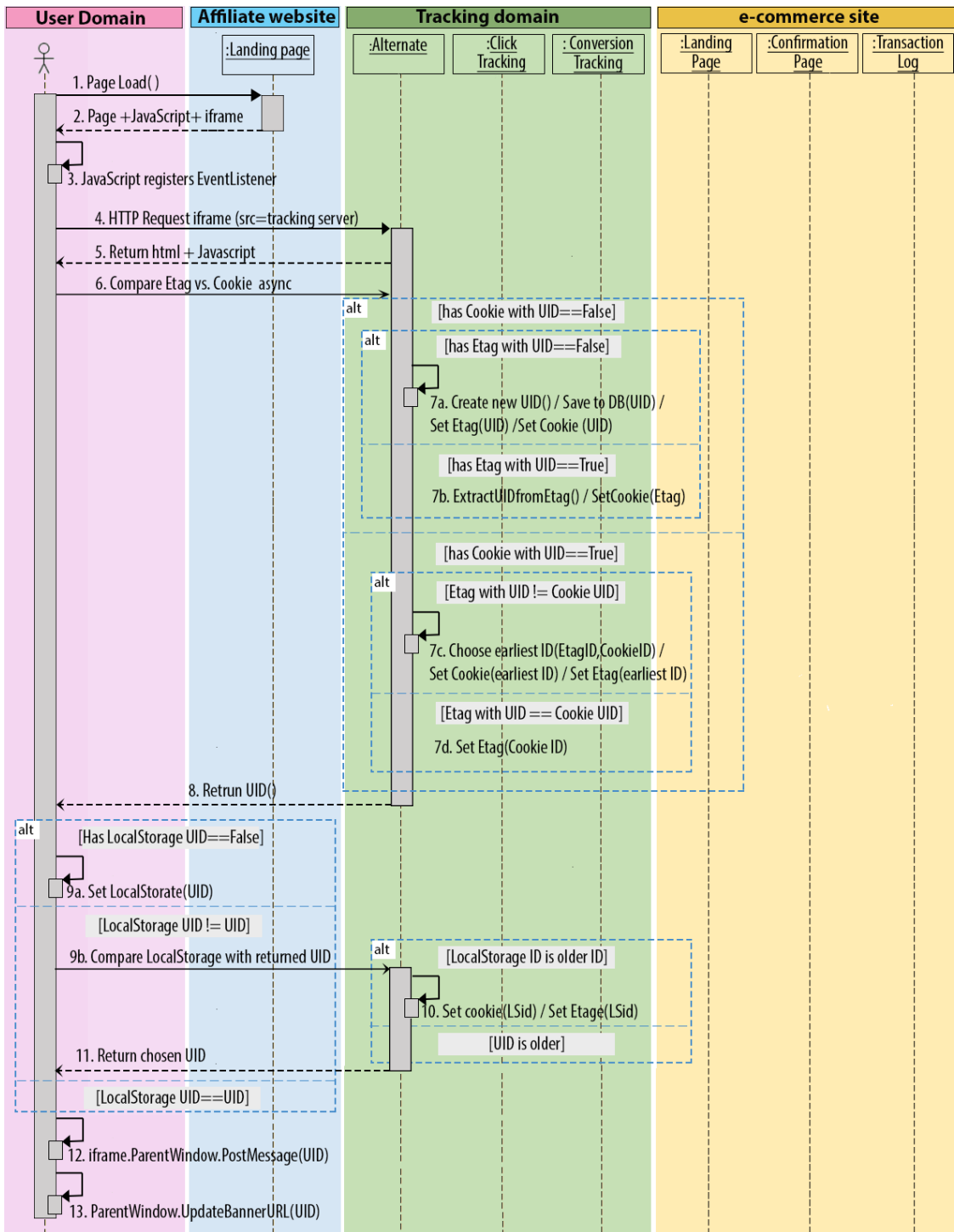


Figure 16: Sequence diagram for page-loading event using robust tracking process

Table 10: First partial list of tracking processes in a robust tracking scenario

Process No.	Process description
1	The user requests the affiliate's landing page by typing the URL in to the browser, clicking a link or using a saved bookmark.
2	The client-browser receives the requested webpage, that contains an invisible iframe. The parent page and the iframe both have JavaScripts attached to them.
3	The JavaScript in parent page registers an EventListener to listen to the messages sent by the iframe.
4	The browser continues loading remaining resources in parent HTML page, by next sending for the source of the iframe. The source HTML page of the iframe is not located at the affiliate domain, but at the tracking domain. A cross-site request is sent to the tracking server.
5	The tracking domain returns an HTML page with a JavaScript embedded in it. The hidden Iframe loads the HTML, but it remains invisible to the user. The embedded JavaScript is next executed.
6	With this process, we start finding the UID, and synchronizing it across all tracking vectors. The JavaScript within iframe makes an asynchronous request to the tracking server's UID synchronizing URL. This HTTP request is accompanied by the HTTP cookie and the ETag, if they exist.
7a	If neither exists, a new UID is created, saved to Database and a new ETag and a HTTP cookie is created with the UID value. The return value is set to this UID
7b	If no cookie is found but has an ETag available, then a new cookie is created with the UID found in the ETag. The UID is set as the return value
7c	If an ETag and HTTP cookie are both found but the values are different, it points at a previous tracking error. Hence, we correct it by choosing the older of the two UIDs by querying the UID database. The ETag and HTTP cookie values are then synchronized with the chosen UID, and set as return value
7d	If a cookie and an ETag found, and they both have the same UID value, this UID value will be set as the return value
8	The UID is returned to the browser. During following processes, the returned UID will be compared with the UID stored in Local Storage, to finally determine the final UID
9a	If the Local storage does not contain a UID, the returned UID is stored in Local storage.
9b	If the Local storage has a UID but the value is different to the returned UID, it makes another roundtrip to compare and determine, which of the two was issued first. The older will be considered the correct UID.
10	If the UID found in Local storage is older, then a new HTTP cookie is set with the UID from Local storage. As this URL is different to the URL used in process 6 to set the ETag, a new ETag will not be set. It is deferred to next tracking session.
11	The chosen UID is returned
12	The JavaScript within the iframe that received the UID, posts it as a message to the parent window
13	Parent window updates the Banner URL with the UID of the visitor

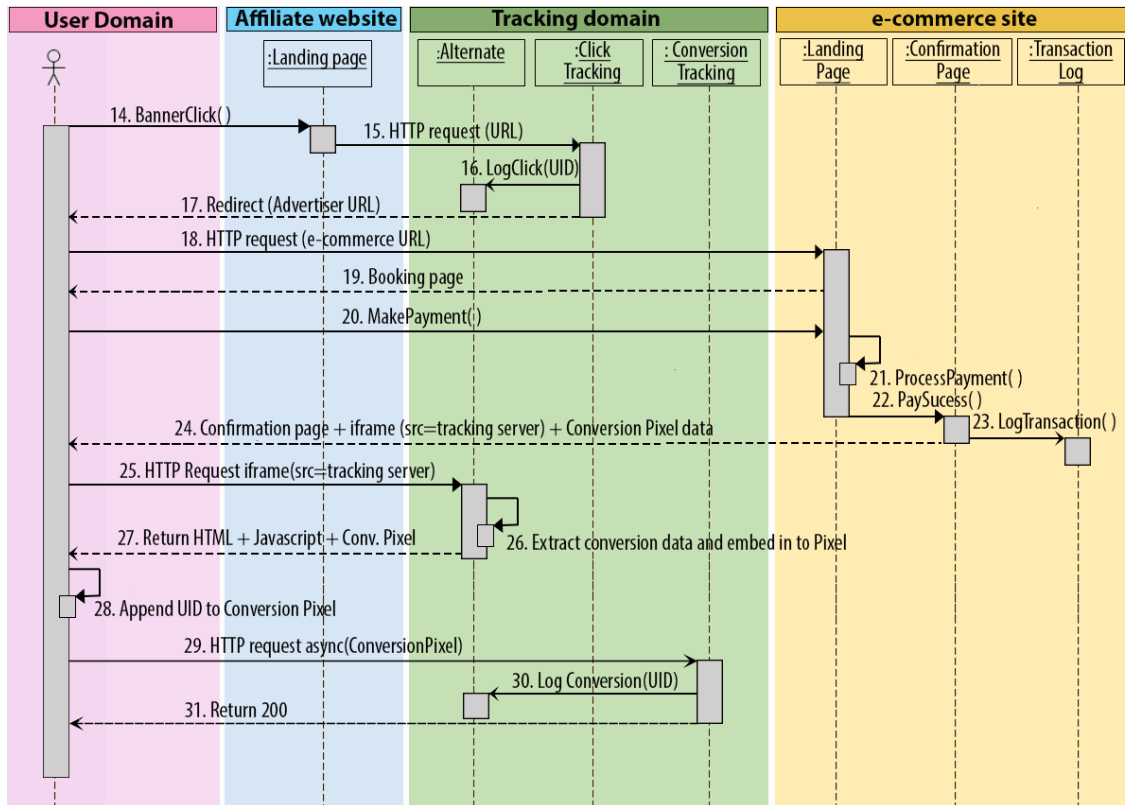


Figure 17: Sequence diagram for click- and conversion tracking using robust tracking process

Table 11: Second partial list of tracking processes in a robust tracking scenario

Process No.	Process description
14	The User clicks the banner, which causes the browser to send an HTTP request to the Tracking server's click-tracking URL, which is the source of the banner URL.
15	The click-tracking process, extracts the UID, Affiliate ID and Advertiser ID from the parameters list of the HTTP request URL
16	An asynchronous process registers a new click-action in the database, against the user.
17	The tracking server returns a 302-redirect response code with the URL denoted by the advertiser ID
18	The browser sends a new HTTP request to the advertiser's URL returned by the tracking server
19	The landing page of the e-commerce site (advertiser) is returned to the browser
20	Usual user interactions, such as browsing the products and adding them to the shopping cart are not shown as they are not relevant to the tracking process and to minimize the clutter. Instead, the user enters a total price and clicks "pay now" button, emulating a payment action
21	The payment is processed and confirmed.
22	Payment process on e-commerce server generates a confirmation page, with an iframe. The iframe has an attached JavaScript and within its URL, transaction data required to create a "conversion-Pixel" as parameters. They include total price, receipt number and advertiser ID.
23	An asynchronous process at the e-commerce server records a new transaction within its database, without blocking the main conversion main process of generating tracking <i>Pixel</i> and the confirmation.
24	E-commerce server returns a confirmation page, showing payment data, and a conversion <i>Pixel</i> data within the URL of a hidden iframe.
25	Browser requests the HTML page, which is the source of the iframe from tracking server.
26	The UID will be read from the accompanying HTTP cookie, and together with transaction data provided by the e-commerce site, we create a conversion <i>Pixel</i> . During initial tracking process that takes place at page loading event, we used ETags, Local storage and cookies to verify and further synchronise the UID. But during conversion tracking process, we assume that those three UID stores are remaining synchronised, therefore would only use the UID value within the cookie, and in the next process, we supplement with a verification of Local storage, but save the extra roundtrips needed for ETag verification.
27	The HTML source for the hidden iframe is returned with a JavaScript and transaction and UID data as parameters of the conversion tracking URL as a conversion- <i>Pixel</i> .
28	The JavaScript within iframe retrieves the UID from Local storage, confirms UID sent by the last process.
29	The JavaScript then executes an asynchronous GET request of the conversion- <i>Pixel</i> , to the conversion-tracking URL.
30	The conversion logging process uses the UID, and Advertiser ID embedded in the conversion <i>Pixel</i> to find the matching click-event from the click tracking table. The affiliate ID, stored with the click-data allows the tracking process to calculate

	the commission amount based on the transaction data, and causes the affiliate to earn the rightful commission for promoting the web-traffic.
31	The tracking server returns a 200 (OK) status code to the browser and completes the interaction.

4.2 Artefacts relating to privacy models

An AM network topology was selected for the experiments carried out in this research study, as such topology includes all cross-domain tracking features, that can easily be modified to suit different test scenarios and use cases. The said network topology comprising of Virtual Machines (described in section 3.3.2) enabled in adding multiple server domains of any category and expanding the network to experiment the effects under a larger network, effortlessly.

Some of the privacy intrusion related experiments that were carried out in the test network AMNSTE2 (described in section 3.4) were then ported to the *Public-AMNSTE* platform. The tracking services were hosted at <https://cnx.ictresearch.co.nz>.

In contrast to the “robustness improvement-based” experiments described above in section 4.1, where only a non-PII based simple UID was used for tracking users, privacy related tracking artefacts were developed to gather as much PII of users, that are available to the web server, within each given use case. Five different categories of use cases were broadly defined, based on their information seeking behaviour and thus resulting privacy intrusiveness, which are discussed in section 6.3.2. The amount of PII and the level of details of a person depends on the technical capabilities within those different use case categories. Privacy-related experiments attempt to demonstrate the granularity of PII sourced at each level of category, to support the categorisation of the privacy model presented in section 6.3.2. The privacy-related tracking artefacts representing different scenarios described in the following subsections were used to capture PII data of the users within each scenario.

A *Pixel* was placed in every webpage that was intended to be tracked. When the visitor loads a tracked page on the browser, the *Pixel* triggers a connection to the tracking URL endpoint at

<https://cnx.ictresearch.co.nz>, during which, the tracking server records a visit against the UID, and web URL of the page visited. *Tracking Pixel* can use any of the resource requests that were used in previous experiments. This experiment has used a request for a CSS file as the resource that triggers code snippet named *Tracking Pixel*.

4.2.1 Single domain tracking

In single domain tracking experiments, the *tracking Pixel* was placed anywhere within the main webpage. If using a CSS or JavaScript file resource request to trigger the tracking process, the tracking *Pixel* can be placed in the header. For example,

```
<head>  
  
<link href="https://cnx.ictresearch.co.nz" rel="stylesheet" >  
  
</head>
```

If an iframe or an image or any other multimedia file request is used as the tracking *Pixel*, it can be placed anywhere in the body, with height and width set to zero *Pixel* (or one *Pixel*).

```
<img src=https://cnx.ictresearch.co.nz width="0" height="0" />
```

Even if the tracked website or the tracking domain has cross-domain restrictions, or Content Security Policy (CSP) restrictions that dictates which content can be loaded from what domains, it does not have an impact on the tracking techniques demonstrated here. Such restrictions are imposed by the browser, not to load content from unapproved sites. But we do not need anything displayed, as they are set to be invisible. The only requirements for tracking technique to work are that the *Tracking Pixel* triggers the HTTP call to the tracking URL and in return the tracking server can set a cookie or an alternative tracking vector (discussed in previous section), with a UID.

4.2.2 Multi-domain tracking

As of now, single-event tracking involved in these privacy-related experiments can use the same techniques for single-domain and multi-domain tracking. The previous section explains how a link to

a CSS file or an iframe or an image in the body with the source set to the tracking URL works within both scenarios. This can be verified using experiments on *Public-AMNSTE* platform. The two important requirements that the *Tracking Pixel* should fulfil is: 1) Trigger a HTTP request to the tracking URL 2) Ability to transmit the UID to the tracking server using any tracking vector. In our privacy related experiments, we have used HTTP cookies, but other alternative tracking vectors discussed in AM experiments can be used too. It cannot be guaranteed, that the technique used in single-domain tracking will continue to function reliably over time. If it fails, a possible solution would be to place the *Tracking Pixel* within an iframe, having iframe's source set to the tracking URL, as follows:

```
<iframe src="https://cnx.ictresearch.co.nz" width="0"
height="0"></iframe>
```

If a different tracking vector than HTTP cookie is used (e.g., Local storage or ETags), then a JavaScript should be used to make an asynchronous HTTP request (AJAX) to the tracking URL. As those tracking vectors are subject to "Same-Origin" restrictions, the JavaScript file should be fetched from the same domain as the tracking URL, as follows:

```
<iframe src="https://cnx.ictresearch.co.nz/Pixel" width="0" height="0">
<script src="https://cnx.ictresearch.co.nz/Pixel/Pixel.js"></script>
</iframe>
```

4.2.3 Business insights gathering

Any website or e-commerce site can log visitor-activity to their site, as in Figure 18, to gather insights on customer interactions with the site. Though the Figure 18 shows tracking records from multiple sites, as the *Tracking Pixel* was placed on all those websites that are under my control, a single e-commerce site can follow the same techniques to track every page within one e-commerce site instead. As evident, this demonstrates a privacy-preserving tracking scenario, as no PII is gathered. A

	NZvistDate	RecID	RQcookieID	ProxyIP	Referrer	Platform	Browser	IsMobile	MobileModel
1	2021-03-26 00:57:...	157788	CXI637523...	118.92.97.172	https://amarasekara.net/	Unknown	Chrome89	1	Linux
2	2021-03-26 00:57:...	157787	CXI637523...	118.92.97.172	https://amarasekara.net/	Unknown	Chrome89	1	Linux
3	2021-03-26 00:48:...	157786	CXI637523...	118.92.97.172	https://slintgl.com/ravi	WinNT	Chrome88	0	Unknown
4	2021-03-25 22:35:...	157785	CXI637522...	66.249.68.11	https://newzealandtravel.net.nz/	Unknown	Chrome89	1	Linux
5	2021-03-25 21:29:...	157784	CXI637522...	66.249.68.15	https://newzealandtravel.net.nz/	Unknown	Chrome89	0	Unknown
6	2021-03-25 20:45:...	157783	CXI637412...	118.92.97.172	https://nztravelguide.org.nz/	WinNT	Chrome88	0	Unknown
7	2021-03-25 20:44:...	157782	CXI637412...	118.92.97.172	https://newzealandtravel.net.nz/	WinNT	Chrome88	0	Unknown
8	2021-03-25 20:44:...	157781	CXI637412...	118.92.97.172	https://newzealandtravel.net.nz/	WinNT	Chrome88	0	Unknown
9	2021-03-25 20:44:...	157780	CXI637412...	118.92.97.172	https://newzealandtravel.net.nz/	WinNT	Chrome88	0	Unknown
10	2021-03-25 20:43:...	157779	CXI637412...	118.92.97.172	https://newzealandtravel.net.nz/	WinNT	Chrome88	0	Unknown
11	2021-03-25 20:42:...	157778	CXI637412...	118.92.97.172	https://nztravelguide.org.nz/	WinNT	Chrome88	0	Unknown
12	2021-03-25 20:34:...	157777	CXI637030...	118.92.97.172	https://nztravelguide.org.nz/	WinNT	Chrome89	0	Unknown
13	2021-03-25 19:34:...	157776	CXI637522...	175.157.75.58	https://slintgl.com/	Unknown	Chrome88	1	Linux
14	2021-03-25 19:33:...	157775	CXI637522...	175.157.75.58	https://slintgl.com/	Unknown	Chrome88	1	Linux
15	2021-03-25 19:31:...	157774	CXI637522...	175.157.75.58	https://slintgl.com/	Unknown	Chrome88	1	Linux
16	2021-03-25 15:21:...	157773	CXI637522...	66.249.68.13	https://amarasekara.net/	Unknown	Chrome89	0	Unknown
17	2021-03-25 15:21:...	157772	CXI637522...	66.249.68.11	https://amarasekara.net/	Unknown	Chrome89	0	Unknown
18	2021-03-25 13:14:...	157771	CXI637412...	118.92.97.172	https://nztravelguide.org.nz/	WinNT	Chrome88	0	Unknown
19	2021-03-25 13:11:...	157770	CXI637412...	118.92.97.172	https://nztravelguide.org.nz/	WinNT	Chrome88	0	Unknown
20	2021-03-25 13:04:...	157769	CXI637412...	118.92.97.172	https://bede.amarasekara.net/	WinNT	Chrome88	0	Unknown
21	2021-03-25 12:57:...	157768	CXI637522...	118.92.97.172	https://amarasekara.net/	WinNT	Chrome88	0	Unknown

Figure 18: Multi-domain visitor tracking results

visitor is only known by the CookieID. A subset that is of interest is shown here, but a researcher can view all the data fields that are captured by interacting with the Public-AMNSTE.

The IP address is not a UID, as the same IP address can be shared by multiple hosts if the ISP uses dynamic IP allocation (DHCP). This table can be merged with a webservice that provides IP data lookups that can show us the country, and often city of the origin IP address, which is an important insight for an e-commerce site to know the geographical location and language of their clients. The *IsMobile* column reveals the visitors who used a mobile device to browser, and if it was a mobile device, then it displays the model and manufacturer. More specific device model can be extracted from the User-Agent column, which is not displayed here, due to lack of space. If the visitor is using a non-mobile device (laptop or a desktop) the Platform column shows the operating system used and the Browser column shows the web browser used. This information can provide insights to the demographics of visitors. When every page in a website is tracked that can further reveal, which pages are more popular and by sorting them on chronological order, the time-gap reveals where customers spent most time. The data can be queried to view all interactions by a specific user, or specific geographic region or language.

Though a visitor is not usually required to be logged-in to browse products in an e-commerce site, to purchase goods, one needs to create a user account and be logged in. Due to payment and delivery features, this would usually require Name, Address and Phone No.'s, etc. The above privacy preserving tracking table can instantly become a privacy intrusive tracking, when the above UID is mapped to the user-account data. These privacy concerns are discussed in detail by evaluating and analysing, in the next two chapters.

Chapter 5. Evaluation

This chapter evaluates the efficacy and utility of the artefacts that were described in previous chapter 4 (*Artefact Description*), which are the main outputs of this research. The iterative process of artefact design and evaluation led to the final artefact outputs that were described in previous chapter. I also present some of the iterative evaluations that led to the abandonment of the development path, or to change the course of the path of some lines of enquiry. The robustness of the tracking process is measured by the constructs of the tracking construct group, during different scenarios. All the three tracking constructs have two properties; Success and Fail, where success state defines the desired state.

The objective of an efficient, reliable, and robust tracking process is to identify an individual user or an individual browser uniquely on the Internet, over a single browsing session at a minimum, but ideally over a long period of time. If such unique identification spans only a single browsing session, it can still be useful for some tracking use cases, such as AM efforts, where click action and conversion action takes place within the same browsing session or where only tracking single events matter (e.g., CPM, CPC). An ideal scenario would be an identification capability that lasts longer than a browsing session, such as a few days, weeks, or months. The ability to determine such duration by the tracking service, would be the best outcome.

5.1 First cycle

The objective of the initial design-evaluation cycle was to find a set of unique properties within the HTTP request object that could enable a webserver to identify browser uniquely. The HTTP request objects, which are the resource requests sent by a browser to the webserver, when loading an HTML page, was examined during the request handling process on server. The *HttpRequest* object within a .NET development environment exposed hundreds of properties, which appeared to be good

candidates to create a unique signature. The literature review revealed some attributes that have been used by stateless tracking techniques to create a unique signature. Though individual attributes are not unique across browsers, a relatively unique signature can be created by combining data mainly acquired by JavaScripts running on a browser (e.g., Canvas, font properties, etc.). My intention was to look for similar unique properties that might be exposed by the HTTP request object, as the .NET *HttpRequest* exposed such a vast array of properties. A quick view using the debugging techniques within the .NET Integrated Development Environment (IDE) proved that many of the properties were not populated during a HTTP request from a browser. No literature was found that discussed how and under what circumstances those values may be populated.

5.1.1 Using recursion to find UID candidates

As the first step, a module was added to the tracking server application of AMNSTE2 that would save the attributes of the properties of *HttpRequest* object to a database. The intention was to then make HTTP requests from different browsers, operating systems, devices, security protocols (HTTP and HTTPS), etc. to determine which *HttpRequest* properties are always populated, and which might populate only under different circumstances. This would enable in determining which properties can be combined for a reliable unique signature generation. While some properties such as *UserAgent* returned a string that identified the client browser and its version, other properties returned an array of strings (e.g., *AcceptTypes*). Some returned a collection of name/value pairs (e.g., *Form*), while others returned additional objects (e.g., *LogonUserIdentity*) or collection of complex objects such as the *Browser* property. The *Browser* property is of interest as it represents the *browser capabilities* collection, which returns a large collection of other objects relevant to those capabilities. Hence, a new software module was added with the capability to recursively loop through the *HTTP request* object's property values if they contained additional collections of other objects. If the collection consisted of strings, they were concatenated to create a single long string value. When the returned collection contained other objects of different types, and those object types were not known

beforehand, *.NET reflection* was used to determine types at runtime, and recurse those objects to extract the properties and their values. Properties with primitive data types were logged as key/value pairs, while properties that were of object data types were added to a list that was then recursed with depth first traversal technique to a desired degree of depth determined at runtime. The utility is accessible for testing at:

<https://www.ictresearch.co.nz/Research/RQobject/>.

A depth first traversal exposed large number of properties that made up the *HttpRequest* object within the .NET Framework. It was observed that most of the properties were not populated under any combination of hardware, software, or operating system configurations. That led me to inspect the same *HTTP request* object, as it is visible within a PHP environment, which returned far smaller number of *HTTP Request* properties. This utility is publicly accessible at:

<https://www.ictresearch.co.nz/asp/Request.php/>.

The difference in the results convinced me that the *HttpRequest* object provided by .NET framework did not populate all the properties using data sent by the client, instead using local .NET libraries that “guessed” those properties based on the *UserAgent* string sent by the client request.

Nevertheless, it was noticed that some of the properties present in both PHP server variables were nearly identical to the corresponding *Server variables* provided by ASP.NET. The properties that relate to the client, returned by PHP and .NET environments are listed in Table 12.

Table 12: PHP and .NET server variables

PHP	.NET
AUTH_TYPE : NULL	AUTH_TYPE : APPLICATIONCOOKIE
AUTH_USER : NULL	AUTH_USER : BEDE@AMARASEKARA.COM
AUTH_PASSWORD : NULL	AUTH_PASSWORD :
CERT_COOKIE : NULL	CERT_COOKIE : NULL
CERT_FLAGS : NULL	CERT_FLAGS : NULL
CERT_ISSUER : NULL	CERT_ISSUER : NULL
	CERT_KEYSIZE : 256
	CERT_SECRETKEYSIZE : 2048
CERT_SERIALNUMBER : NULL	CERT_SERIALNUMBER : NULL
	CERT_SERVER_ISSUER : C=US, O=Let's Encrypt, CN=Let's Encrypt Authority X3
CERT_SUBJECT : NULL	CERT_SERVER_SUBJECT : CN=ICTRESEARCH.CO.NZ
CONTENT_LENGTH : NULL	CONTENT_LENGTH : 0
CONTENT_TYPE : NULL	CONTENT_TYPE : NULL
HTTPS : ON	HTTPS : ON
HTTPS_KEYSIZE : 256	HTTPS_KEYSIZE : 256
HTTPS_SECRETKEYSIZE : 2048	HTTPS_SECRETKEYSIZE : 2048
HTTPS_SERVER_ISSUER : C=US, O=Let's Encrypt, CN=Let's Encrypt Authority X3	HTTPS_SERVER_ISSUER : C=US, O=Let's Encrypt, CN=Let's Encrypt Authority X3
HTTPS_SERVER_SUBJECT : CN=ICTRESARCH.CO.NZ	HTTPS_SERVER_SUBJECT : CN=ictresearch.co.nz
	HTTP_CACHE_CONTROL : max-age=0
HTTP_CONNECTION : CLOSE	HTTP_CONNECTION : close
HTTP_ACCEPT : TEXT/HTML,APPLICATION/XHTML+XML....	HTTP_ACCEPT : text/html,application/xhtml+xml,application/xml;...
HTTP_ACCEPT_ENCODING: GZIP, DEFLATE, BR	HTTP_ACCEPT_ENCODING : gzip, deflate, br
HTTP_ACCEPT_LANGUAGE : EN-GB,EN:Q=0.9....	HTTP_ACCEPT_LANGUAGE : en-GB,en;q=0.9,de;q=0.8,en-US;q=0.7
HTTP_COOKIE : LONG COOKIE STRING...	HTTP_COOKIE : LONG COOKI STRING.....
HTTP_HOST : www.ictresearch.co.nz	HTTP_HOST : www.ictresearch.co.nz

HTTP_REFERER : HTTPS://WWW.ICTRESEARCH.CO.NZ/	HTTP_REFERER : https://www.ictresearch.co.nz/account/login/?returnUrl=/aspx/request.aspx
HTTP_USER_AGENT: MOZILLA/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit...	HTTP_USER_AGENT : Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML...
HTTP_UPGRADE_INSECURE_REQUESTS : 1	HTTP_UPGRADE_INSECURE_REQUESTS : 1
HTTP_SEC_FETCH_MODE : NAVIGATE	HTTP_SEC_FETCH_MODE : navigate
HTTP_SEC_FETCH_USER : ?1	HTTP_SEC_FETCH_USER : ?1
HTTP_SEC_FETCH_SITE : : SAME-ORIGIN	HTTP_SEC_FETCH_SITE : same-origin
INSTANCE_ID : 589 ??maybe App ID on server	INSTANCE_ID : 589
FCGI_ROLE : RESPONDER	
LOCAL_ADDR : 27.123.28.65	LOCAL_ADDRESS : 27.123.28.65
LOGON-USER :	LOGON_USER : BEDE@AMARASEKARA.COM
QUERY_STRING : NULL	QUERY_STRING : NULL
REMOTE_ADDR : 203.118.180.182	REMOTE_ADDR : 203.118.180.182
REMOTE_HOST - 203.118.180.182	REMOTE_HOST : 203.118.180.182
REMOTE_PORT - 15176	REMOTE_PORT : 15176
REMOTE_USER - NULL	REMOTE_USER : BEDE@AMARASEKARA.COM
REQUEST_METHOD -GET	REQUEST_METHOD : GET
PATH_TRANSLATED : W:\VHOSTS\ICTRESEARCH.CO.NZ\HTTPDOCS\ASP X\REQUEST.PHP	PATH_TRANSLATED : W:\VHOSTS\ICTRESEARCH.CO.NZ\HTTPDOCS\A SPX\REQUEST.ASPX

It was recognized that a few of the properties that were sent by the client browser to the server, together with the collection of headers were useful inputs, lacked the ability to compile a unique signature for each user. Additional data generated by a JavaScript running within the browser and reporting back to the server, as used by stateless tracking techniques discussed in section 2.4 was required. Though stateless tracking techniques have improved in reliability and accuracy, considering the high resource usage, latency, and lesser accuracy than stateful tracking, I decided to maintain the scope of this research to alternative stateful tracking methods.

The popular stateful tracking mechanism, the HTTP cookie is reliable and versatile in *Click-tracking* and in *Conversion-tracking* scenarios. But previous research has found HTTP cookie-based tracking can fail under certain circumstances. Hence, those failure conditions were examined next, to define the construct properties and their measurements for the experiments.

5.2 Tracking failures

Tracking failures are not fraudulent activities, nevertheless it has a negative impact on e-commerce activities. Affiliates can lose their rightfully earned commissions if the tracking system fails. In a cookie-based tracking system, some users may block cookies, causing the browser not to save cookies in their computers. By default, cookies are allowed and saved in a browser. Even if a user has disabled cookies, the browser still uses “Session cookies”, that are discarded at the end of the browsing session, which therefore still allows the tracking system to function, but only if the user who clicked a banner-advertisement, goes on to make a purchase, within the same browsing session, before exiting the browser.

5.2.1 Fail scenarios

Some situations were examined, under which tracking can fail:

1. When the browser cache is cleared, or cookies individually deleted between the click-tracking process and conversion-tracking process.
2. If the “incognito mode” or similar “private” browsing feature offered by most browsers, was used, and does not complete the “conversion” in the same browsing session, instead exits the browser, and starts another session of the browser to continue the purchase. If the user completes the conversion within the same browsing session, then the tracking process will succeed, as incognito mode uses session cookies, which has the lifespan of the session duration.
3. If the cookie has expired when the user returns to complete the purchase.

4. 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.
5. 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. The conversion will not be tracked, as the cookie was placed in a different computer.

5.3 Defining a baseline for Tracking techniques

My next experiment was to determine which tracking techniques discussed in previous literature were still relevant, at the start of this research. Those techniques may then be included in the next stage of this research experiments to improve the robustness of the tracking process. Each of the technologies listed in Table 13, was evaluated against the listed capabilities forming a baseline for assessed technologies.

Table 13: Currency of tracking technologies

Evaluated technology	Tracking capabilities			
	Single event	Multi-event	Single domain	Cross-domain
<i>Flash cookies</i>	Fail	Fail	Fail	Fail
<i>Microsoft Silverlight</i>	Fail	Fail	Fail	Fail
<i>HTTP cookies</i>	Success	Success	Success	Success
<i>HTML5 Local storage</i>	Success	Success	Success	Success
<i>Etag</i>	Success	Success	Success	Success

HTTP cookie which is part of the HTTP protocol, is the only mechanism that was developed for the purpose of maintaining state; tracking is a result of maintaining state (Kristol & Montulli, 1997). None

of the alternative tracking vectors were originally developed for tracking purposes. For example, *Local storage* is a client-side technology to save individual user and web domain specific data locally, *Etag* is a cache management technology to save network bandwidth and latency by caching frequently downloaded resources. *Flash cookies* are a local storage for Adobe Flash content, etc. Hence, technological advances and changes in implementations and specifications of these alternative techniques can change over time in future, which can render them ineffective as tracking vectors. Table 13 shows a summary of findings. Adobe Flash cookies and Microsoft Silverlight failed to demonstrate any tracking capability and were found to be obsolete during our experiments. Other technologies tested were found still capable of being used as tracking vectors, for specific tracking requirements and to a specific degree.

The experiments further revealed that the use of “*Flash cookies*” (Flash Local Storage) to “respawn” or re-instantiate deleted cookies, discussed by Soltani, et al., (2010) has been curtailed since version update 10.3 of Adobe Flash player. Neither was *HTML5 Local storage* able to respawn deleted cookies as described by Ayenson et al., (2011), as the browsers that have been tested so far, have in the meantime blocked that functionality. But currently, I have found that Local storage and ETags can still help to make tracking technology more robust in other ways. ETags were capable of re-spawning deleted HTTP cookies and restore Local storage, on most occasions. The test results show that ETags are more robust against tracking failures discussed above.

Successful tracking vectors from Table 13, were then subjected to a set of experiments presented in Table 14 below, to evaluate how well they perform under conditions that failed HTTP cookies, which are listed in sub section 5.2.1. Results of the experiments are presented below. They verify the efficacy of those alternative tracking vectors, when used individually and the technique we propose, by combining multiple vectors is presented as “robust tracking”

Table 14: List of experiments to check efficacy of tracking vectors

Test 1	Loading an HTML page or clicking a banner advertisement on an AM page causes a visit to be accurately registered on the tracking server.
Test 2	The ability for payment confirmation pages of e-commerce sites to accurately and reliably transmit the affiliate identifier, transaction ID and total price to the tracking server. Capabilities of test 1 and test 2 together encompass the tracking process needed for an AM network.
Test 3	Ability to simultaneously maintain visitor identity between two windows of the same browser.
Test 4	Ability to simultaneously maintain visitor identity between two tabs of the same browser.
Test 5	Despite the “private browsing” mode of a browser, the tracking server has ability to identify a user with a previously saved identifier instead of recording them as a new user (Fail scenario 2)
Test 6	Ability to identify a visitor uniquely when using different browsers on the same device (Fail scenario 4)
Test 7	Ability to continue to identify a visitor even after the browser cookies are deleted (Fail scenario 1)
Test 8	Ability to continue to identify a visitor even after the browser cache has been deleted (Fail scenario 1)
Test 9	Non-expiring unique identifier (Fail scenario 3)

5.4 Evaluation of tracking capabilities tests

After evaluating the *End of Life* (EOL) announcements and the evaluations of currency of tracking technologies mentioned in previous sub-section, Flash cookies aka. Adobe LSO (Adobe, 2020), Microsoft Silverlight (Microsoft, 2020) and Web SQL (Hickson, 2010), those technologies were excluded from experiments. It was determined, Indexed DB behaved like the HTML5 local storage, hence HTML5 Local storage and ETags were chosen to be further tested for their suitability as alternative tracking vectors.

5.4.1 Evaluating test results

A set of experiments were designed to test the efficacy of chosen tracking vectors under different scenarios. Table 14 lists the experiments, that measure the dichotomic *tracking capability* property which relates to the success vs. failure of the measured construct as a tracking vector, within the experimented application context. The test scenarios include single-event and multi-event tracking instances as well as scenarios under which HTTP cookie-based tracking failed.

The set of experiments listed in Table 14 made up a major part of this research, which also warranted most of the “development time”. The network was configured as mentioned in chapter 3, (Methodology). As part of this research, three types of application software were developed for the three different types of domains used in the experiments, i.e., an Affiliate, E-commerce, and a Tracking domain. The tracking domain contained the bulk of software development effort, as it is at the heart of the entire tracking process.

The experiment environment was setup to simulate an *Affiliate Marketing Network*, as such network incorporates all the technological capabilities that are expected to be carried out during this research. As a multi-domain tracking environment, an AMN can be used for cross-domain tracking experiments; Click-tracking represents a single-event tracking scenario while conversion tracking represents a multi-event tracking scenario, that can happen across multiple browsing sessions over a longer period.

The summary of nine test results in Table 15 shows, that “super cookie” concept (Ayenson et al., 2011; Soltani et al., 2010) discussed in previous research, has no relevance at present. Super cookie concept was not one specific tracking vector, but a combination of technologies, when used in tandem would result in an indestructible tracking solution that can easily re-spawn deleted cookies. As they employ technologies that were not originally meant for tracking purpose, despite being effective at that time, later versions of those technologies have made them partially ineffective. Nevertheless, the partial

successes can still be utilised to create the tracking solutions more robust. Table 15 shows the status of current relevance.

Table 15: Results of efficacy of alternative technologies as tracking methods

	Cookies	Local Storage	ETags
Test 1	Success.	Success	Success
Test 2	Success	Success	Success
Test 3	Success	Success	Success
Test 4	Success	Success	Success
Test 5	Fail	Fail	Fail
Test 6	Fail	Fail	Fail
Test 7	Fail	Fail	Success
Test 8	Fail	Success	Fail
Test 9	Fail	Success	Success

All the experiments in this research were carried out using the multi-domain test environment AMNSTE2 that was created as part of this research, which is described in chapter 3. It simulates the Internet, using the same network protocols and technologies, and its purpose was to provide me unhindered access to the network and to servers, without triggering security alerts during experiments. As AMNSTE2 is an internal network and is not accessible on the Internet, at the end of this research, part of the network topology was re-created using multiple domains that are publicly accessible on the Internet, enabling researchers and industry practitioners to check the functionality described in this research. As I describe the experiments below, the entry point URLs to the publicly available experiments are provided alongside, where available.

Test 1:

Loading a page or clicking a banner on any of the tracked pages of the e-marketing domains causes a visit to be accurately registered on the tracking server.

This experiment demonstrates single-event tracking capability within a cross-domain context. Many tracking use cases require only this capability, which is relatively easy to implement.

Some of the use cases are:

- A website that tracks interactions of a user navigating through webpages during a browsing session within a single domain
- A third-party tracking company which tracks a user across multiple websites
- Display advertising (CPM) which only needs to track each instance of a banner advertisement being displayed on a webpage, but not any further interactions with user
- Some business analytics providers, who provide customer demographics related to visitors to a website can use the tracking techniques discussed in this experiment

Different variations of this test have been made available on the Public-AMNSTE platform, to verify the success of the techniques presented. Entry points to the test pages are publicly accessible at <https://nztravelguide.org.nz/> and <https://newzealandtravel.net.nz/>. Each page contains information, on how to carry out the tests and how to access the results page.

Within an AM context, the above test scenario is represented by a click-tracking instance, where a user clicks on a banner advertisement, and the source URL of the banner image is set to the tracking URL, which causes a single tracking event to be triggered on the tracking server. In a display advertising scenario, this tracking event can be implemented using a JavaScript that executes a cross-domain HTTP request to the tracking server. If it is needed to be executed within a *same-site* request, it can be executed when the JavaScript is run within an iframe, whose source is set to the tracking domain. Without using JavaScript, it can also be executed as an HTTP resource request for one of the many kinds of resources, such as a hidden image, CSS file, an image defined within a CSS file or any other type of multi-media file. I have demonstrated at the above Public-AMNSTE platform how such tracking event can also be implemented, without any user action, unbeknown to the user in the background, by executing the resource request code within the page's load event.

Experiments using different resource types to cause single-event tracking are demonstrated at <https://nztravelguide.org.nz/test.html> and <https://newzealandtravel.net.nz/test.html>. During each experiment, the interactions between the browser and the webserver can be observed using developer tools of the browser and the result can be examined within the *Click Results* table at <https://connex.net.nz/track/results/>. The *Click-Results* table in Figure 19 shows results in descending order. A tracking success is reflected by a new tracking record with a new *Tracking ID* and current date and time in UTC in *Click date* column. *VisitorID* that starts with *ET* are results of tracking using ETags. Prefix *LS* denotes Local storage-based tracking technique. The proposed robust tracking technique-based results do not have a prefix.

As the test results in Table 15 show, all three tracking vectors, viz. HTTP cookies, HTML5 Local storage and ETags succeeded in performing this tracking test accurately.

	TrackID	clickdtm	SessionID	VisitorID	RQparams	AffID	AdvID	OfferID
1	11805	2021-03-27 00:40...	toyjsupb3dtemgmico1kjzo	637395624524301817	F10A105	10	1	5
2	11804	2021-03-27 00:33...	h3yti3jnco3hjd0miwqz5jr4	LS637395624524301817	F20A105	20	1	5
3	11803	2021-03-27 00:31...	xkpw04uuc5jnqvpmekj3gh	LS637395624524301817	F10A105	10	1	5
4	11802	2021-03-27 00:16...	tjmsycom0hipbda1jpl1ipy5	ET637524449507487555	F10A105	10	1	5
5	11801	2021-03-26 23:28...	a5n5wt4qn1dyltyapnnqwrfl	ET637524408966997279	F10A105	10	1	5
6	11800	2021-03-26 23:15...	aptrqjypr44gxoukrcaas0l	637395624524301817	F20A1020	20	1	20
7	11799	2021-03-26 22:47...	bbgx5ufe0gk0tqatwa2kgeya	637395624524301817	F10A205	10	2	5
8	11798	2021-03-26 22:39...	hon122nlse10fnnqiheo15s	ET637524408966997279	F20A105	20	1	5
9	11797	2021-03-26 22:36...	ahhu0fvh2xxqhs4gmpxsbfvf	ET637524408966997279	F20A105	20	1	5
10	11796	2021-03-26 21:36...	bripdx2g5dkhoiukxi2rcalh	ET637524378427763464	F20A105	20	1	5
11	11795	2021-03-26 21:33...	cajjouqiqow4hh51mo0yifqq	ET637524378427763464	F20A105	20	1	5
12	11794	2021-03-26 21:30...	0w4l4hfjbqne1fqhjpjpeblf	ET637524378091357094	F20A105	20	1	5
13	11793	2021-03-25 22:06...	2g1zsb5zhqfco4b5nyqqpea0	637523535829987742	F10A205	10	2	5
14	11792	2021-03-25 22:06...	bdmbrwykvkdcwr3qfvxbqv	637523535813269154	F10A2010	10	2	10
15	11791	2021-03-25 22:06...	qm2t5mikotrqsocp5p50hjn	637523535803268969	F10A304	10	3	4
16	11790	2021-03-25 22:06...	5xibrs4dq254fi3o5utysegc	637523535795144540	F10A2015	10	2	15
17	11789	2021-03-25 22:06...	mvoeefcu24ewmzo2nwd5gzrl	637523535777332059	F10A101	10	1	1

Figure 19: Click results in descending order

Test 2:

The ability for payment confirmation pages of e-commerce sites to accurately and reliably transmit the affiliate identifier and total price of items purchased to the tracking server. Capabilities of test 1 and test 2 together encompass the tracking process needed for an AM network.

This experiment is represented by a conversion-tracking event within an AM context. Unlike test 1, this test involves tracking a user beyond the single event, during further interactions. For example, within an AM context under CPA advertising model, clicking on a banner advertisement, and arriving at an e-commerce site does not earn any payment for the affiliate. The visitor must make a purchase, so that a percentage of the purchase value will be paid as a commission. A purchase can take place within the same browsing session, else the user might return at a later date to complete the purchase action. Therefore, this experiment tests the capability of the tracking technology to identify a user even at a later date. This experiment can be carried out on the publicly available test site, at <https://nztravelguide.org.nz> or <https://newzealandtravel.net.nz>, as in the previous experiment, but then continuing to emulate a purchase action in the e-commerce site. The result of a successful tracking event can be evaluated at <https://connex.net.nz/track/results/>, where the first action (click-

Conversion Results

Conversion ID	Conversion Date	CookielD	Tracking ID	Transaction ID	Advertiser ID	Affiliate ID	Offer ID	Amount	Click count
163	2/24/2021 1:46:12 PM	LS637498177342285691	11717	293	1	20	5	60.0000	1
162	2/24/2021 1:44:54 PM	LS637498177342285691	11717	292	1	20	5	30.0000	1
161	2/24/2021 1:43:52 PM	ET637498178139317338	11716	291	1	20	5	20.0000	1
160	2/24/2021 12:33:13 PM	LS637492607592324442	11673	289	1	20	5	10.0000	1
159	2/24/2021 12:29:09 PM	LS637492607592324442	11673	275	1	20	5	20.0000	1
158	2/24/2021 12:29:09 PM	LS637492607592324442	11673	276	1	20	5	60.0000	1
157	2/24/2021 11:06:11 AM	LS637395624524301817	11646	287	1	20	5	40.0000	5
156	2/19/2021 12:25:59 AM	LS637395624524301817	11646	285	1	20	5	10.0000	5
155	2/19/2021 12:22:04 AM	ET637493375906623423	11681	284	1	20	5	60.0000	2
154	2/19/2021 12:20:24 AM	ET637493375906623423	11680	283	1	20	5	50.0000	1

Figure 20: Last 10 conversion results in descending order

action) should be recorded in the *Click Results* table and the corresponding purchase action should be recorded at the *Conversion Results* table shown in Figure 20 .The results from Table 15 shows that all three tracking vectors can perform this tracking capability successfully.

Test 3 & 4:

Ability to simultaneously maintain visitor identity between two windows of the same browser.

&

Ability to simultaneously maintain visitor identity between two tabs within the same window of a browser.

Within any of the tracking use cases discussed above, a user may choose to have more than one browser tab or multiple windows of the same browser opened at any time. During this experiment, we test if our tracking vectors can accurately recognize the user, when interacting through any of the open tabs or open windows. This experiment can be executed using the publicly available test site, by carrying out the above two experiments within multiple opened tabs and observing the results.

Results in Table 15 shows that all three tracking vectors are capable of tracking a user successfully under both test scenarios.

Test 5:

Despite the “private browsing” mode of a browser, the tracking server has ability to identify a user with a previously saved identifier instead of recording them as a new user (Fail scenario 2)

This experiment tests tracking capability within an “incognito window”, during private browsing sessions. Different browsers name this browsing mode as “Private mode”, “Incognito mode or as “InPrivate” mode; this browsing mode is explicitly made available to the users in situations where the user interactions should be isolated from all previous sessions and any future sessions. No content related to visited sites, browsing history, passwords or cookies should be saved on the device. Hence,

the specification recommends browser manufacturers to treat persistent storage in the same manner as HTTP cookies within a *Private browsing* environment, thus restricting access to it, and avoiding saving application state to the storage. It also recommends browser manufactures to delete data from persistent storage related a specific domain or all domains, as per the user’s intent, when a user is deleting HTTP cookies (Alabbas & Bell, 2018, 2021; Olejnik, 2019). Since the introduction of HTML5 and associated local storage solutions more than a decade ago, different browsers implemented the above recommendations to different degree of compliance. By 2021, all major versions of browsers now treat local storage in the same way as HTTP cookies.

As shown in Table 15 all tracking vectors failed to track a visitor under this test scenario. The importance and relevancy of this failure is further discussed in the next chapter.

Test 6:

Ability to identify a visitor uniquely when using different browsers within the same device (Fail Scenario 4)

Browsers do not share cookies, Local storage nor browser caches between browsers. Therefore, when using a different browser within the same computer, a user appears as a new user to a website. As the experiment results in Table 15 shows, all three tracking vectors failed to track a user under this test scenario. Now dysfunctional techniques such as *Flash cookies* and *Silverlight* have been successful under this scenario, as Adobe Flash was a plug-in used by all major browsers, before the advent of HTML5, which shared its storage and other software components among all client browsers within a physical computer. That allowed websites to access the UIDs saved in Adobe Local storage from any browser. Like the conclusion of previous test, current development trends avoid using plug-in support for most such multimedia extensions, instead relying on functionality provided with HTML protocol since HTML version 5.

Test 7:

*Ability to continue to identify a visitor even after the browser cookies are deleted
(Fail scenario 1)*

This test demonstrates a major improvement in robustness of the tracking process. A user may click a banner advertisement and arrive at an e-commerce site, thereby receiving an UID for tracking, but postpones the purchase event to a later date. In between, if the user deletes the cookie cache of the browser, the user can not be identified with the previous click event, and the affiliate will lose her rightfully earned commission, which is a major drawback of the HTTP cookie-based tracking process. The experiment shows, if the cookie-based tracking process is supplemented with the alternative techniques proposed in this research, the tracking process can retain the UID in above scenario. As of now, when cookies are deleted, the ETag tracking vector still retains the value, thereby tracking the user successfully. I also have demonstrated that the deleted HTTP cookie and the HTML5 storage can also be respawned in the same process, making the tracking technology even more robust.

The purpose of ETags is to cache web resources to save bandwidth and expedite loading a page on the browser. Therefore, as of now, when cookies are deleted ETags do not get simultaneously deleted though Local storage gets cleared.

Test 8:

*Ability to continue to identify a visitor even after the browser cache has been
deleted (Fail scenario 1)*

In contrast to the previous *cookie-cache* clearing test which cleared all the cookies cached by the browser, this *browser-cache* clearing test is about clearing everything including cached pages and all other resources. Though it is more destructive, we found the way the cache clearing facility is implemented in all the major browsers, added another layer of protection to the robustness of the HTTP cookie-based tracking system. Facility for clearing cookie-cache is provided within easy reach on the user interface, but browser-cache clearance is not offered to the user with the same ease. A user requires to activate *Developer Tools* screen, which is usually beyond usual user-access. Therefore, ETags that are associated with browser cache increases the robustness of the tracking process.

Test 9:

Non-expiring unique identifier (Fail scenario 3)

HTTP cookies have a lifespan which is set at the time of creation. Tracking requirements over long periods can use the advantage of Local storage and ETags, that do not expire. When ETags are used for its originally intended purpose of caching resources, the specification recommends returning a *304 Not Modified* status code, if the ETag has not changed. But in these experiments, I demonstrated how to use ETags as a tracking vector, in which case the server uses the ETag as an UID. Instead of returning a *304 Not Modified* status code, which represent *No Change* to the original resource, therefore causes the browser to abandon further processing, the tracking technique sets the same UID as ETag on every request, so that the browser will continue to process the response. This also ensures the ETag continues to retain its UID value.

The results of the above experiments as shown in Table 15 demonstrate that the alternative tracking techniques in this experiment can increase the versatility of the cookie-based tracking technique.

5.5 Privacy intrusion simulations

The privacy related experiments were designed based on *Information Seeking Behaviour* (ISB) of the tracking use cases. When tracking is used purely as an underlying technology, ISB is at minimum, and it complies with non-privacy invasive tracking techniques. Other use cases described in section 2.3 have different levels of ISB underlying their main purpose of tracking users online. Most of the negative perceptions of user-tracking originate from this category of use cases, some being synonymous with stalking.

In the following subsections, we present data that was captured using applications that fall within different use case categories described in section 6.3.2. The captured data within each use case category demonstrate different levels of PII data available to the webserver applications at each level,

and I discuss within each subsection how those PII information can be further enriched with additional information.

5.5.1 Tracking as an underlying technology

This is the least privacy invasive category as discussed in the privacy model under section 6.3.2. Many e-commerce activities such as web traffic generation models (CPA, CPC, etc.) need the capability to know which channels promoted the sale, so that appropriate commissions can be paid to the channel operators. As shown in Figures 19, 20 and 21 the user is only identified with a UID, which does not reveal any PII of the user. Located at the bottom of the privacy intrusiveness model, this use case is a privacy preserving tracking model.

Privacy related experiments discussed here were carried out as part of cookie, Local storage and ETag based tracking experiments. With the data from those experiments shown in Figure 21, it is demonstrated that tracking can be done reliably and, in a privacy-preserving manner without using or exposing any PII information of users. During these experiments, individual users are not identified, instead the server generated UID which is based on the date and time of the visit, represents a browser. No PII of the user was used nor was necessary for the tracking process. In general usage, a computer user routinely uses one browser on a single machine, thus a UID issued for a browser is

	TrackID	clickdtm	SessionID	VisitorID	RQparams	AffID	AdvID	OfferID
1	11805	2021-03-27 00:40...	toysupb3dtemgmico1kjzo	637395624524301817	F10A105	10	1	5
2	11804	2021-03-27 00:33...	h3yti3jnco3hjd0miwqz5jr4	LS637395624524301817	F20A105	20	1	5
3	11803	2021-03-27 00:31...	xkpw04uuc5rjnqvpfmekj3gh	LS637395624524301817	F10A105	10	1	5
4	11802	2021-03-27 00:16...	tjmsycom0hipbda1jpl1ipy5	ET637524449507487555	F10A105	10	1	5
5	11801	2021-03-26 23:28...	a5n5wt4qn1dyltyapnnqwmf	ET637524408966997279	F10A105	10	1	5
6	11800	2021-03-26 23:15...	aptrqjypr44gxoukrcaas0l	637395624524301817	F20A1020	20	1	20
7	11799	2021-03-26 22:47...	bbgx5ufe0gk0tqatwa2kgeya	637395624524301817	F10A205	10	2	5
8	11798	2021-03-26 22:39...	hon122nlse0frnqiheo15s	ET637524408966997279	F20A105	20	1	5
9	11797	2021-03-26 22:36...	ahhu0fvh2xxqhs4gmpxsbfvf	ET637524408966997279	F20A105	20	1	5
10	11796	2021-03-26 21:36...	bripdx2g5dkhoiukxi2rcahv	ET637524378427763464	F20A105	20	1	5
11	11795	2021-03-26 21:33...	cajjouqjqow4hh51mo0yifqq	ET637524378427763464	F20A105	20	1	5
12	11794	2021-03-26 21:30...	0w4l4hfjbbqne1fqhvjvpeblf	ET637524378091357094	F20A105	20	1	5
13	11793	2021-03-25 22:06...	2g1zsb5zhqfco4b5nyqapea0	637523535829987742	F10A205	10	2	5
14	11792	2021-03-25 22:06...	bdmbrwykvvkdccwr3qfvxbqv	637523535813269154	F10A2010	10	2	10
15	11791	2021-03-25 22:06...	qm2t5mikotrqsocpt5p50hjn	637523535803268969	F10A304	10	3	4
16	11790	2021-03-25 22:06...	5xibrsm4dq254fi3o5utygsc	637523535795144540	F10A2015	10	2	15
17	11789	2021-03-25 22:06...	mvoeecfu24ewmzo2nwd5gzrl	637523535777332059	F10A101	10	1	1

Figure 21: Results of Click-Tracking in descending order

synonymous with a user. But during my experiments, multiple popular browsers were used, therefore multiple UIDs displayed in Figure 21, can be different browsers on the same computer, used by a single user. The “VisitorID” column contains the UID. The first three records belong to the same user, the UID being represented by the numeric part of the Visitor ID. As this table holds records of different experiments that used different tracking vectors, the Visitor IDs that contain only a numeric value as in first record originated In *Robust Tracking* experiment described in section 4.1.4. When the same numeric UID is prepended with *LS* as in second and third records, they represent the results of Local storage experiment described in section 4.1.2. The UIDs prepended with *ET* are results of ETag experiments. Records 1, 2, 3, 6 and 7 belongs to the same UID and therefore represent a single browser, but the time stamp and more importantly the “*SessionID*” shows that our experiments using different tracking vectors have successfully tracked a user over different browsing sessions, and the different “*AffID*” (Affiliate ID) among those records indicate, that they were successfully tracked across different unrelated websites. As the table data shows, there is sufficient information to verify which affiliate generated the web traffic, and to which e-commerce site (advertiser ID) and which advertised offer, as different offers can attract different fees or commissions. Based on that information, fees, or commissions due to each affiliate can be calculated. The user is only known by the UID and many months or even years of interactions at different sites can thereby be monitored, without causing a privacy intrusion.

A single instance of a tracking server (connex.net.nz) continued to track visitor interactions across all participating domains. The network was expanded by adding multiple instances of e-commerce and affiliate servers which enabled me to simulate a real-world networks of multiple e-commerce sites subscribing to one tracking service provider.

5.5.2 Tracking for information gathering

All tracking use cases that displayed ISB at different levels were grouped under this category. It starts with a least privacy-invasive scenario such as an e-commerce site gathering business insights locally.

They then extend towards most privacy-intrusive scenarios such as search engines and browser manufacturers, whose primary intention is to gather as much user related data.

Business insights gathering experiment

Starting with the least privacy intruding ISB experiment, the previous experiment was expanded, simply by adding a few more data fields, that are available to us through the HTTP request object. It is done without the requirement of a Log-in or account creation, thereby still without gathering any PII data of a user. This use case represents an e-commerce company gathering useful visitor insights, without personally identifying a user. Figure 22 shows visit data of six visitors to the site.

Each page of the website contained a tracking *Pixel*. Therefore, when a visitor navigates from page to page, a new tracking record is saved in the database, giving the e-commerce site the capability to gather important insights relating to visitor interaction. In this experiment we can determine that the six different *CookieIDs* represent six different users. In previous experiments, where IP addresses were not tracked, six different *CookieIDs* can mean either six different users or a single user using different browsers, as browsers do not share cookies and each new browser appears to the webserver as a separate entity. When we add the IP address to the mix, we can see these are six different users as

Recid	Date Time UTC	CookieID	SessionID	RemoteHost	Referrer	Platform	Browser
1	2019-12-20 22:58:41...	CXI637030517261265490	4mz3yaogl5lcyqmQjuv1a1wr	203.118.180.182	https://slintgl.lk/?page_id=21	WinNT	Chrome79
2	2784 2019-12-20 23:01:24...	CXI637030517261265490	4mz3yaogl5lcyqmQjuv1a1wr	203.118.180.182	https://slintgl.lk/?page_id=21	WinNT	Chrome79
3	2785 2019-12-20 23:01:51...	CXI637030517261265490	4mz3yaogl5lcyqmQjuv1a1wr	203.118.180.182	https://slintgl.lk/	WinNT	Chrome79
4	2786 2019-12-20 23:02:26...	CXI637030517261265490	4mz3yaogl5lcyqmQjuv1a1wr	203.118.180.182	https://slintgl.lk/	WinNT	Chrome79
5	2787 2019-12-20 23:02:46...	CXI637030517261265490	4mz3yaogl5lcyqmQjuv1a1wr	203.118.180.182	https://slintgl.lk/	WinNT	Chrome79
6	2790 2019-12-21 01:28:41...	CXI637088878317446995	14dl34qfn3nyncofoeqwtmj4o	116.206.247.249	https://slintgl.lk/	WinNT	Firefox71
7	2791 2019-12-21 02:10:12...	CXI637088878317446995	14dl34qfn3nyncofoeqwtmj4o	116.206.247.249	https://slintgl.lk/?p=272	WinNT	Firefox71
8	2792 2019-12-21 02:11:41...	CXI637088878317446995	14dl34qfn3nyncofoeqwtmj4o	116.206.247.249	https://slintgl.lk/?page_id=21	WinNT	Firefox71
9	2793 2019-12-21 02:11:52...	CXI637088878317446995	14dl34qfn3nyncofoeqwtmj4o	116.206.247.249	https://slintgl.lk/?page_id=21	WinNT	Firefox71
10	2794 2019-12-21 02:11:57...	CXI637088878317446995	14dl34qfn3nyncofoeqwtmj4o	116.206.247.249	https://slintgl.lk/?page_id=111	WinNT	Firefox71
11	2795 2019-12-21 02:12:47...	CXI637088878317446995	14dl34qfn3nyncofoeqwtmj4o	116.206.247.249	https://slintgl.lk/?page_id=111	WinNT	Firefox71
12	2796 2019-12-21 02:20:42...	CXI637088878317446995	14dl34qfn3nyncofoeqwtmj4o	116.206.247.249	https://slintgl.lk/?page_id=111	WinNT	Firefox71
13	2797 2019-12-21 02:20:48...	CXI637088878317446995	14dl34qfn3nyncofoeqwtmj4o	116.206.247.249	https://slintgl.lk/?p=272	WinNT	Firefox71
14	2798 2019-12-21 02:22:14...	CXI637088878317446995	14dl34qfn3nyncofoeqwtmj4o	116.206.247.249	https://slintgl.lk/?p=272	WinNT	Firefox71
15	2799 2019-12-21 03:14:36...	CXI637030517261265490	4mz3yaogl5lcyqmQjuv1a1wr	203.118.180.182	https://slintgl.lk/	WinNT	Chrome79
16	2802 2019-12-21 07:07:09...	CXI637125376295502976	czooowztigbbcmhgtplrs0cx2	175.157.42.165	https://slintgl.lk/	WinNT	Chrome79
17	2804 2019-12-21 19:38:49...	CXI637125827296042641	hvaqbtgnsoqxs2cvisl3r0	171.13.14.5	https://slintgl.lk/	WinNT	Chrome79
18	2811 2019-12-22 16:30:45...	CXI637091130944020580	p5pqygtclzjhkq5f3l2g5q2vp	112.134.131.166	https://slintgl.lk/	WinNT	Chrome78
19	2812 2019-12-22 16:30:46...	CXI637091130944020580	nkafnjp5ti1j22o0vc0iktah	112.134.131.166	https://slintgl.lk/	WinNT	Chrome79
20	2813 2019-12-22 16:32:51...	CXI637091130944020580	nkafnjp5ti1j22o0vc0iktah	112.134.131.166	https://slintgl.lk/?page_id=20	WinNT	Chrome79
21	2814 2019-12-22 16:38:09...	CXI637091130944020580	nkafnjp5ti1j22o0vc0iktah	112.134.131.166	https://slintgl.lk/?page_id=20	WinNT	Chrome79

Figure 22: E-commerce site gathering business insights locally

they have six different IP addresses, which give away their geographical locations too. An IP look up reveals the following information displayed in Table 16.

Table 16: IP address information

IP address	Internet Service	City	Country
203.118.180.182	Vodafone NZ	Auckland	New Zealand
116.206.247.249	Dialog-LK	Colombo	Sri Lanka
175.157.42.165	Dialog	Colombo	Sri Lanka
171.13.14.5	China Telecom Henan	Zhengzhou	China
112.134.131.166	Sri Lanka Telecom	Colombo	Sri Lanka

The *referrer* field shows the sequence of the pages visited and the time gap between two entries indicates the time spent on each page, thereby indicating which pages are most popular and how long users spend on those pages. Some of the columns that are not shown in Figure 22 due to lack of space, such as “*Is Mobile*”, “*Mobile Model*”, “*User-Agent*” and “*Browser Capabilities*”, reveal additional details. “*Is Mobile*” column reveals if the visitor is using a mobile device to browse, and “*Mobile model*” reveals the model and manufacturer. More specific device model data can be extracted from the *User-Agent* column. If the visitor is using a non-mobile device (laptop or a desktop) the *Platform* column shows the operating system used, and the *Browser* column shows the web browser used. Though this experiment demonstrates an active ISB by the e-commerce site, which is gathering more information than required to function as a technological necessity, it does not gather PII. Hence, it can be seen as a balanced effort in insights gathering while preserving the privacy of user.

5.5.3 Tracking by third-party business analytics services

Though an e-commerce site can gather business insights locally, as demonstrated in previous subsection, most enterprises use a third-party service provider, mainly due to two different reasons.

1. The e-commerce site does not have the technical expertise to implement tracking and information gathering expertise. There are cost-free services readily available, who offer premium services at an additional cost.
2. The e-commerce site is interested in gathering more insights than what is available within a local insight gathering endeavour, which has only visibility over interactions within the local site. When hundreds or thousands of such websites have subscribed to services of a large analytics service provider, such provider has visibility over user-interactions across any of those subscribed sites. It can provide insights on customer interests based on what products were perused or purchased at other monitored websites. For example, if a customer has purchased an air ticket to a holiday destination at one e-commerce site, other e-commerce sites that sell accommodation, leisure activities, travel accessories, tours and transports will be interested in those marketing leads.

The experiment described in section 4.2.4 was used to demonstrate the use case of a third-party business analytics service. In real-world this use case is represented by AMNs who provide business analytics services in addition to third-party tracking services, as well as services such as Google's *Universal Analytics* service. They provide services to many e-commerce sites, and each subscribed site has a *Tracking Pixel* of the specific service provider embedded in the webpages. When a visitor visits any of the monitored sites, visitor-interactions are being tracked in the tracking database. A visitor carries the same UID across all the websites that are being monitored by the tracking service, hence their interaction across all those monitored websites can be compiled in to one dossier of an individual visitor's interactions, over time.

Figure 23 shows the results of the experiment, where individual landing pages are tracked. The "CookieID" shows interactions of four different users across multiple sites. The displayed tracking results are from the *Public-AMNSTE* platform, tracked by the tracking service located at

<https://cnx.ictresearch.co.nz/>, which provides tracking services to all the websites seen in *Referrer* column.

Here we combine the IP address, which is unique to a single user during a browsing session. IP addresses are usually not unique over a longer duration, depending on the IP address leasing period of the DHCP server, as the IP addresses are re-allocated after specific interval.

But the geographical location of the user is revealed by the IP address, which can be matched with browser language to reveal the possible nationality or ethnicity of the visitor. It is an important information for marketing and strategic planning to know the composition of visitors to an e-commerce site. On the other hand, it can be used for specific fraud prevention purposes too.

Database queries can further summarize or extract specific information on visitors such as, the first visit, frequency of subsequent visits as well as successful monetary outcomes, total purchase values, purchase per visit ratios and similar business insights from tracking data as shown in Figure 20. As the visitor moves between other e-commerce or affiliate sites, that have subscribed to the same tracking service, the “referrer” header of the HTTP request revealed the previous domain name, while the unique identifier remains the same across all the visited domains. Any products that were perused at

	NZvistDate	RecID	RQcookieID	ProxyIP	Referrer	Platform	Browser	IsMobile	MobileModel
1	2021-03-26 00:57:...	157788	CXI637523...	118.92.97.172	https://amarasekara.net/	Unknown	Chrome89	1	Linux
2	2021-03-26 00:57:...	157787	CXI637523...	118.92.97.172	https://amarasekara.net/	Unknown	Chrome89	1	Linux
3	2021-03-26 00:48:...	157786	CXI637523...	118.92.97.172	https://slintgl.com/ravi	WinNT	Chrome88	0	Unknown
4	2021-03-25 22:35:...	157785	CXI637522...	66.249.68.11	https://newzealandtravel.net.nz/	Unknown	Chrome89	1	Linux
5	2021-03-25 21:29:...	157784	CXI637522...	66.249.68.15	https://newzealandtravel.net.nz/	Unknown	Chrome89	0	Unknown
6	2021-03-25 20:45:...	157783	CXI637412...	118.92.97.172	https://nztravelguide.org.nz/	WinNT	Chrome88	0	Unknown
7	2021-03-25 20:44:...	157782	CXI637412...	118.92.97.172	https://newzealandtravel.net.nz/	WinNT	Chrome88	0	Unknown
8	2021-03-25 20:44:...	157781	CXI637412...	118.92.97.172	https://newzealandtravel.net.nz/	WinNT	Chrome88	0	Unknown
9	2021-03-25 20:44:...	157780	CXI637412...	118.92.97.172	https://newzealandtravel.net.nz/	WinNT	Chrome88	0	Unknown
10	2021-03-25 20:43:...	157779	CXI637412...	118.92.97.172	https://newzealandtravel.net.nz/	WinNT	Chrome88	0	Unknown
11	2021-03-25 20:42:...	157778	CXI637412...	118.92.97.172	https://nztravelguide.org.nz/	WinNT	Chrome88	0	Unknown
12	2021-03-25 20:34:...	157777	CXI637030...	118.92.97.172	https://nztravelguide.org.nz/	WinNT	Chrome89	0	Unknown
13	2021-03-25 19:34:...	157776	CXI637522...	175.157.75.58	https://slintgl.com/	Unknown	Chrome88	1	Linux
14	2021-03-25 19:33:...	157775	CXI637522...	175.157.75.58	https://slintgl.com/	Unknown	Chrome88	1	Linux
15	2021-03-25 19:31:...	157774	CXI637522...	175.157.75.58	https://slintgl.com/	Unknown	Chrome88	1	Linux
16	2021-03-25 15:21:...	157773	CXI637522...	66.249.68.13	https://amarasekara.net/	Unknown	Chrome89	0	Unknown
17	2021-03-25 15:21:...	157772	CXI637522...	66.249.68.11	https://amarasekara.net/	Unknown	Chrome89	0	Unknown
18	2021-03-25 13:14:...	157771	CXI637412...	118.92.97.172	https://nztravelguide.org.nz/	WinNT	Chrome88	0	Unknown
19	2021-03-25 13:11:...	157770	CXI637412...	118.92.97.172	https://nztravelguide.org.nz/	WinNT	Chrome88	0	Unknown
20	2021-03-25 13:04:...	157769	CXI637412...	118.92.97.172	https://bede.amarasekara.net/	WinNT	Chrome88	0	Unknown
21	2021-03-25 12:57:...	157768	CXI637522...	118.92.97.172	https://amarasekara.net/	WinNT	Chrome88	0	Unknown

Figure 23: Multi-domain visitor tracking results

other domains give away the current purchase interests of the visitor. The knowledge of non-purchase of a product at one site, can be sold to the next site as a premium lead such as the “remarketing” leads provided by many such services. A query for each session, that does not include the page URL of the payment page returns all customers who perused products but did not make a purchase. This group of records can be further queried based on the time spent on individual product pages, showing which products caught the attention of the user, most. If the same product or category was perused at multiple e-commerce sites, such customers can be promoted as strong “remarketing” leads to other e-commerce sites, that sells similar products. At this level of tracking, despite knowing the approximate location, language and buying habits, the visitor is only identified by a UID, without any PII.

As described in section 4.2.4 we examined three different use cases within this category, which represented three different real-world scenarios. It exposed that even under a third-party AM network or under a business analytics service, a user can have three levels of privacy intrusions and ISB’s based on the use case:

- I) non-PII, when only a UID is used (as in above experiment)
- II) limited PII usage based on PII provided by a user at the time of creating a local login account at an e-commerce site (as in next experiment use case)
- III) highest level of exposure by combining the local user account with social media data (as in last use case in this section).

As described in section 4.2.4, we extended our simulation experiment to the second level of privacy-intrusion, by adding a user-account creation and log-in feature to some e-commerce sites. This is a regular feature in real-world e-commerce websites, as customers need to securely log-in to order and pay for purchases. Customer’s PII such as name and contact details and delivery addresses are usually required to complete a transaction.

That led to the first level of personal privacy intrusion, as that enabled the tracking service to combine the anonymous user-persona that was well developed in the previous non-PII experiments, with a real person who is identifiable with an email address. Names, addresses or any other information could be gathered in this process, depending on the motivation to lead a user to provide additional data in exchange of services provided.

We further extended this experiment to the third and highest level of privacy-intrusion by adding the facility of an external authentication service. Instead of a local account that uses a user name and password, by offering the log-in facility with Facebook, Google, Microsoft, Twitter and similar authentication providers, we can combine UID used in previous experiments with the user name and the provide key of the external login account to reveal the OSN profile name of a user as shown in Figure 24. This enables further enrichment of the tracked persona with information that appear in OSN platforms, thus causing a much higher level of privacy intrusion. PII data in the table in Figure 24 was anonymised for privacy protection.

CookieID	ProviderKey	Provider	CreateDTM	UserName	Email
CXI637030517261265490	4788801775498222	Facebook	10/02/2021, 11:42:50	fdantbdkkdera	hfdera.kdantba@hdail.hom
CXI637088878317446995	404487182298825895720	Google	12/11/2020, 02:52:56	b.H.h.dvNvgaAAhPUHAMd	bhhdhhfdanh@nmata.hdm
CXI637091130944020580	899845441844423	Facebook	05/11/2020, 12:31:21	hdddahfnribkna	fahddhkndibkna@nmaia.hsm
CXI637091997826622303	887213998399524	Facebook	29/08/2020, 11:30:31	hdhnnngdbvantha	hahninif1222@ndaia.bdm
CXI637094532283773569	40881458308355495	Facebook	09/08/2020, 04:50:47	dvhanbJadakthfdava	dahdjaganb@yahoo.hov
CXI637125376295502976	422942124799074	Facebook	17/07/2020, 02:12:12	hddhanbvtgthunadafk	hdfad2dabk@dahyo.com
CXI637125827296042641	925174020524744	Facebook	15/07/2020, 12:49:41	dvntbdajahafdhk	dadahafdbkajnth@hvaia.hsm
CXI637126927123681305	405044824817442	Facebook	15/07/2020, 04:40:39	dvhdvAdavnnnaSaahanf	dahdvsaaahanf@gmanl.bom
CXI637127247733668505	40452774918774822	Facebook	14/07/2020, 05:48:51	dgdnaVimvsbdahasinnhf	dgdnavimasb@yahoo.hom
CXI637128397472387750	40844485400824189	Facebook	22/06/2020, 06:50:32	ddhvfekaWeedakson	dahdfekaweedakson93@yahoo.hov
CXI637128403204105586	40881024737870499	Facebook	20/06/2020, 01:03:54	nbtadthtaAnfdyinnhe	nhtaahthaa@ndail.hdm
CXI637128403208065769	40420004271808408	Facebook	16/06/2020, 04:37:15	bhdhnadannadafk	bhahna8219@dahyo.cod
CXI637128403234496995	444484744854197884289	Google	16/06/2020, 12:46:22	dgdntbvvyafshy	dgdntbvay67@hmaia.hsm
CXI637128403245437475	404750749809918787805	Google	15/06/2020, 07:36:15	fdvaahdkmabhahnra	hfadaah@nmanl.bdm
CXI637128403259458122	404851538894392435378	Google	15/06/2020, 07:31:29	bhdhnanaChandana	bhahnana99321@ndana.cdm
CXI637128406020482208	400901842809754028949	Google	14/06/2020, 09:39:11	ddvhanbfgaatuhna	dadhanbtsgds1975@hmata.hom
CXI637128462715938693	444421988949082208454	Google	14/06/2020, 05:21:44	dbdshnfahaaahpaththi	dbashnfaha@gdaia.hsm
CXI637128462828463754	4513347700803835	Facebook	14/06/2020, 05:11:38	dvvndaalhiad	davndaalhiad@nvail.bsm
CXI637128463126073896	405158438387288705452	Google	13/06/2020, 12:03:29	hgddbndfKuaadathnk	hgddbndfahk@ndaia.cdm

Figure 24: UID merged with social media account data

5.6 CDN exposure

We created a service endpoint on tracking server to serve a JavaScript library simulating the common use of JavaScript libraries from public CDNs. Web pages were created within the Dev domain that had links to those JavaScript libraries within their headers. Some pages were setup to use “Local Storage” as tracking technology in place of HTTP-cookies (Laperdrix et al., 2016).

CDNs are popular among web developers to reduce network latency. It is also common practice to link to most of the popular JavaScript libraries, CSS files and font files through CDNs. A compromised JavaScript file can provide control and access to sensitive data within a page, or in “Local Storage” and user inputs. Our tests were able to steal the visitor IDs from Local Storage, hidden fields on forms, change DOM elements, etc. Other static content providing CDNs can be used to stuff cookies, as discussed in cookie stuffing fraud in AM (Amarasekara & Mathrani, 2015; Sanchez-Rola et al., 2021).

Sanchez-Rola et al. (2021) discovered in their study using large-scale fine-grained crawler a convoluted network of actors, who created and shared tracking cookies and reciprocally exchanged content in webpages, often without the explicit knowledge or consent of the website owners. The shared content from CDNs contain JavaScript files that can extract information and content from each website that uses CDN resources, and cookies that allow cross-domain tracking between them.

5.7 Tracking vector summary - utility, efficacy, and ease of use

HTTP cookie is the de-facto tracking mechanism, which is part of the HTTP state management protocol. Setting a cookie with a UID in a client-browser and sending the cookie back and forth between the server and the client-browser with each HTTP request/response happens as a part of the HTTP protocol. When the server needs to identify the user, it simply reads the UID off the cookie. In contrast, other alternative tracking vectors that we experimented with, during this research, need additional steps to pass the UID to the server. For example, the HTTP cookie-based tracking process described under 4.1.1 starts with the process “*The user clicks the banner advertisement*”, while Local storage and ETag based tracking processes starts banner advertisement clicking at process 11. Cookie based tracking process has the least number of processes involved, while all other alternative tracking methods described here have many other background processes happening during the page-load event and has many more processes to complete a click-tracking and a conversion-tracking process.

5.7.1 Local Storage

Newest versions of Local storage specifications recommend browsers to treat persistent storage as cookies, which has taken some time to implement (Hickson, 2021). Our experiments confirmed that the behaviour of the local storage repository has many similarities to the HTTP cookie cache in a browser, in latest versions of all popular browsers. In 2017, when the first experiments of this research were carried out, Local storage retained its values during cookie deletions in most browser implementations, and therefore was selected as an alternative tracking vector. As of 2021, the Local storage data get cleared, when cookies are deleted, and they are very much aligned with HTTP cookie behaviour. In contrary to previous research findings discussed in literature review, our experiments revealed that Local storage cannot be used to respawn cookies.

Unlike HTTP cookie the Local storage is a client-side technology, which is only accessible to the browser through JavaScript. It is a mechanism to save user-specific data locally on user's computer. Therefore, it is not accessible to the server. For us to repurpose it, to be used as a tracking vector, we need to make it available to the server during each user interaction (e.g., when user clicks on a banner advertisement). This was accomplished by using a JavaScript to read the UID from Local storage during the page load event and appending it to all URLs as a parameter. For example, a *Click-Pixel* on a banner advertisement on an affiliate's webpage has a URL that points at click-tracking URL on the tracking server, with affiliate ID (f=10), advertiser ID (a=2), and other optional parameters such as campaign ID, offer ID (o=5), etc. appended to the URL as parameters, as follows:

```
<div id="divLeftCol">
  <a href="https://connex.net.nz/Track/click/?a=2&f=10&o=5" id="banner">
    
  </a>
</div>
```

The above parameters of affiliate and advertiser IDs are known at design time, which are static, therefore can be delivered as static content during page-load event. At the time the browser loads the page, the JavaScript can dynamically append the user's UID as v=123456, assuming UID=123456.

That solved the problem of passing the UID to the server, which worked well in single domain tracking scenarios. During multi-domain based tracking, (e.g., during a CPA session), the click-tracking occurs in affiliate domain, while conversion-tracking happens at e-commerce domain. The click-tracking could not be matched with the conversion-tracking record, as the UID extracted from Local storage was different during affiliate website access and during e-commerce site access.

That leads to another important technical consideration, which is the placement of JavaScript file. Just as each HTTP cookie has a “Domain” field, that controls which domain owns and has access to the cookie, the local storage too is unique to each domain. A webpage may contain numerous HTML resources such as images, videos, iframes, script files, etc. many could be from different origins, having fetched from different domains. Each resource delivery could send a cookie; therefore, a single web page can contain tens or hundreds of cookies from different domains. Similarly, when Local storage is used, each domain is assigned its own local storage reserved for that domain. One domain cannot access the cookie or the dedicated local storage of another domain.

Consider a multi-domain tracking scenario, where SiteA.com and SiteB.com are using *connex.net.nz* as tracking server. Both sites have embedded JavaScripts in their respective webpages to request a UID from the tracking server for new visitors, and both sites save them in their local storages. When a user visits SiteA.com, the UID generated by tracking server will be saved in user’s local storage, dedicated to SiteA.com. When the same visitor goes to SiteB.com, the previous UID is not available to SiteB.com, as it was saved in the Local storage dedicated to SiteA.com. Therefore, each website can successfully identify the visitor locally, within that site using the issued UID, but not globally, when visiting other websites. For the tracking server, they appear as two different people with two UIDs. The idea of multi-domain tracking is to identify a person with a global UID across all domains, so that the tracking server can create a comprehensive history of all customer interactions on different websites. It was possible to overcome this “Same-Origin” restriction by adding an iframe to each webpage and configuring the *source* of iframe to the tracking server domain. The JavaScript, that

sends and receives the UID from tracking server was placed within the iframe's source page, instead of the parent page. As every tracked website has an iframe, whose *source* is tracking domain, the UID becomes globally accessible among all tracked websites. As the iframe is purely for tracking purpose, the size can be set to 0 or 1 *Pixels* in size, making it invisible. The UID is saved to and later retrieved from the Local storage that is dedicated to the tracking domain, which is accessible to all the webpages with an iframe, of all tracked domains. This phenomenon can be observed when using the *Public-AMNSTE* platform for experiments on Local storage at:

<https://nztravelguide.org.nz/Alternate/uselocalstorage.html>

<https://newzealandtravel.net.nz/Alternate/uselocalstorage.html>

The above test pages display the UID received by the JavaScript file on parent page and that from iframe, both for information. The tracking results page of the tracking domain displays click-actions and conversion-actions of the user, at both unrelated web domains above, can identify the user with a single global UID. The results page is at:

<https://connex.net.nz/track/results/>

It is important to note that when Local storage is used as a tracking vector, the UID stored in Local storage is never available to the server on the first HTTP request of a browsing session. Which means, the first page returned cannot be customised for the user, unlike when using a cookie, where the UID can be extracted, and web contents can be customised with each web request. Similar customisation when using Local storage as tracking vector, can be achieved by using AJAX (asynchronous requests). The page is first sent with static content together with an embedded JavaScript file. The script then runs on client browser, extracts the UID from the Local storage, and sends asynchronously back to the server. The server will then create personalised content based on the UID and return to the browser, which will then populate the page. All the above processes happen asynchronously in the background.

In experiments discussed here, the UID was obtained and appended to the *click-Pixels* in the above manner.

After finding the solution to obtain a global UID, the “Same-Origin” restrictions prevented the iframe updating the *click-Pixel* URLs on the parent page. The parent page and the iframe cannot access each other directly due to different origins. Message posting technique was used to communicate the UID from iframe to the parent page. A JavaScript on parent page registers an *EventListener* to listen to the “*postMessage*” from iframe. The iframe extracts the UID and posts it to the parent page using a “*postMessage*”, which was then appended to the URL of the *click-Pixel* on parent page.

5.7.2 ETag

The function of ETag (Entity Tag) as defined in HTTP specification is to send a conditional request to the server as a cache management mechanism. Instead, the Sequence diagram on Figure 15 demonstrates how a complete user-interaction across multiple domains can be tracked without using HTTP cookies, instead using ETags. The usage is slightly tweaked in contrast to the specification recommendations, to accomplish the use of ETag as a tracking vector. The tracking server software is configured to generate a UID for each new visitor and send it as an ETag to the browser. Unlike when using Local storage, sending the UID as an ETag back to server with each subsequent HTTP request, is looked after by the browser, as part of the HTTP protocol.

When used as a cache validation mechanism, the server sends a unique ETag, with each resource request that a browser may cache. The client browser caches the resource with the ETag. On a subsequent request for the same resource, the client sends a header "If-none-match: <"ETag">" to the server. If the ETag sent by the browser does not match the current ETag for the same resource on server, the updated version of the resource will be sent to the browser to cache, together with the new ETag. If ETags do match, it indicates that the resource has not changed, therefore the server sends

a "304-Not Modified" header, thus saving the bandwidth and latency when downloading the same resource repeatedly.

In the modified usage of ETag as a tracking vector, we use it for tracking a user instead of cache validation. To avoid "*Same-Origin*" restriction discussed in previous subsection, an iframe whose source is set to the tracking domain, has been used in every page that needs to be tracked. Within the iframe a "tracking *Pixel*" with a JavaScript is used to connect with the tracking server to obtain a UID for new visitors. Each request URL to obtain or to verify the ETag, must be always identical across all requests from all domains. Any variation in protocol (http or https), parameters, etc. will cause the server to create a new ETag, thus failing the tracking process. The sequence diagram on Figure 15 shows the extra steps taken, so that no parameters (e.g., Total sale value, transaction ID, advertiser ID, etc) are appended to the URL, unlike when using HTTP cookies or Local storage. In the modified usage, though the server wants to retain the UID unchanged as ETag, the tracking server does not return a "*304-Not Modified*" result, instead sets the same ETag again. Sending a 304 result terminates further processing on the browser, as it indicates that the resource has not changed; not setting the ETag again would essentially delete the cached ETag and its cached resource.

If a client browser makes an HTTP-request to the tracking URL without the "If-none-match" header, it indicates a new user who doesn't have a Unique tracking ID; hence a new one is sent as the ETag. If the user browses any webpage from any domain, that contains the above *Tracking Pixel*, the HTTP request to the tracking URL will always accompany the "If-none-match" header with the given ETag, by which the current visitor can be identified across domains.

In usual tracking usage the iframe will be made invisible, setting the size to zero *Pixels*. But on the *Public-AMNSTE* test environment's ETag demonstration pages the iframe has been kept intentionally visible. This enables the researchers verify the need to execute the ETag related JavaScript code from

within the iframe, instead of executing from main parent page. The ETag demonstration pages are located at:

<https://newzealandtravel.net.nz/Alternate/ETag.html>

<https://nztravelguide.org.nz/Alternate/ETag.html>

In this chapter, I have demonstrated how HTML5 Local storage and ETags can be successfully used as tracking vectors, by slightly tweaking the usage recommended in specifications. The results of individual experiments presented in this chapter and the summary of the efficacy of alternative technologies shown in Table 15 provide proof that alternative technologies can improve the robustness of the HTTP cookie-based tracking process. In next chapter, findings of the above experiments are generalised outside of this current application context.

Chapter 6. Discussion

It was discussed and demonstrated in previous chapters that tracking is a technological necessity in a stateless ecosystem, such as the Internet. This research aims at improving the robustness of the underlying tracking and state management techniques in a privacy-preserving manner and disseminating the knowledge thus acquired to update current knowledge base on efficacy and currency of previously discussed tracking techniques. This was achieved by capturing the complexity of the research problem, within three research goals.

6.1 Research goal 1

The task at hand was to update the knowledge base, on the current status of multiple tracking vectors discussed in previous literature, through experiments and evaluation. Our results (Table 13) have shown while *Flash cookie* concept and other third-party web storages such as those used by *Microsoft Silverlight* have become obsolete, while browsers are moving to implement storage and multimedia capabilities introduced with HTML5, thus avoiding security vulnerabilities involved with third-party browser plug-ins. Through experiments, it was found that HTML5 Local storage and ETags are still usable as tracking vectors. Nevertheless, the experiment results in Table 15 show, they are not indestructible anymore, in contrary to the findings of Ayenson et al. (2011) and Soltani et al. (2010), as browsers have been implementing protocol recommendations to align storage and privacy preserving behaviour of web storages with HTTP cookie behaviour (Barth & Berkeley, 2011). As ETag is not a web storage mechanism, it has the capability to retain the value under circumstances where other tracking vectors fail. Therefore, as shown in Table 15, ETags can be used to increase the robustness of the tracking process.

6.2 Research goal 2

HTML5 Local storage and ETags were chosen as the two alternative tracking vectors to experiment further, how the HTTP cookie-based tracking system's robustness can be improved. There successful implementation within the experimented AM context, was discussed in previous chapter. Here the findings are further generalised, as it may be applicable to a wider context.

In developing an online tracking solution, two approaches are available, based on the level of integration between the tracking solution and e-commerce software, as internal tracking solution (ITS) or external tracking solution (ETS). The internal or external prefix does not represent the physical location of the tracking server, but how far it is coupled with the e-commerce application software. The tracking technique differ in each implementation and strengths and weaknesses need to be assessed based on implementation needs, as discussed below.

6.2.1 Internal tracking solution

An ITS can be used to track only within one domain, similar to the single-domain tracking scenarios discussed previously, but this categorisation is based on software integration. An ITS is convenient for an e-commerce site to track user-interactions within its own domain. While handling each resource request, the web application can identify the browser with a previously set UID, thus reducing an extra roundtrip to a tracking server. It avoids information leaks to third parties, as discussed later in the chapter.

The disadvantage is two tightly coupled web and tracking applications. It also requires ongoing technical capabilities of an in-house software development team to keep up with evolving tracking technologies and associated fraud preventions strategies, which is only affordable to very large e-commerce practitioners. Though such application can capture customer interaction within the enterprise domain efficiently, most marketing teams demand wider range of customer demographics

available through third-party business analytics providers, based on customer interactions beyond their own domains.

6.2.2 External tracking solution

With this architecture, the e-commerce software that generates the webpage, with which a visitor interacts is completely de-coupled from the tracking software artefacts, and they can reside in either within the same domain or in two different domains. In case of a third-party service provider, they reside in two different domains. But an enterprise that wish to carry out tracking tasks internally, as described in ITS sub-section above, may decide to de-couple the tracking artefacts to avoid the disadvantages of tightly coupled applications, but follow the same techniques as and ETS, while keeping the service within the same domain.

This architecture requires a *Tracking Pixel* to be placed on each tracked webpage. While the client-browser loads the webpage from e-commerce webserver, the embedded tracking *Pixel* will cause an HTTP request to be sent to the tracking domain, giving tracking server the chance to track the event. ITS solutions discussed in previous sub-section does not require a *Tracking Pixel*, as any resource request within a page, can be used to track the user.

Any resource can be a *Tracking Pixel* provided it can make an HTTP request and receive a HTTP cookie with the response (in case of Cookie-based tracking, else a HTTP request that send any other tracking vector). An iframe, whose source is set to the tracking URL is a common Tracking Pixel. An image or any other multimedia file, a JavaScript or CSS files are other such options. The External-AMNSTE2 platform demonstrates how different resources can be used as *Tracking Pixels*, which can be investigated interactively at <https://nztravelguide.org.nz/test.html>.

Third-party tracking and business analytics services providers eliminate the need for in-house technical expertise for SMEs, by offering a simplified implementation strategy. A typical *Tracking Pixel* is a small code block that causes a HTTP request to the tracking URL as below:


```
<div id="TrackingPixel">
  <a href="https://connex.net.nz/Track/click/?a=2&f=10&o=5" id="banner">
    
  </a>
</div>
```

When a new e-commerce site signs up with a tracking or business analytics service provider, the only technology implementation required by the new client is to cut and past the above code block anywhere within the body section of the webpages, that need to be tracked.

6.2.3 Single-event tracking

The tracking need of some e-commerce activities, is limited to a single event, such as: (a) capturing visited website URLs or individual page URLs of a user, (b) an advertisement appearing on user's screen (CPM), (c) a user clicking on an advertisement (click-tracking or CPC), (d) a user signing up for a membership, email list, a petition, etc. Usually, such single events are associated with user traffic generation. The website, search engine or the entity that promoted that event to the user, gets rewarded for that event. While many such individual events within a single browsing session may need to be captured by the tracking server, none of those events may require validation through a corresponding event that may happen within a different domain at a different time. Each event is a single independent and completed event, under single-event tracking scenarios. In such scenarios, the underlying tracking technology used is simpler, easier to implement, less error-prone and there is a wide selection of tracking methods to use.

CPM and CPC models are single-event tracking scenarios that do not need to track the visitor beyond the current tracking session. If the same visitor clicks on the same banner advertisement (CPC) or visits the same webpage that displays a banner advertisement (CPM), many times over many browsing sessions, each such event is a sperate legitimate event, that needs to be captured as a new event that would earn a fee. Hence, such tracking implementation does not even need the capability of persisting a UID within an HTTP cookie or within any other tracking vector. The *Tracking Pixel* merely needs to trigger a HTTP resource request to the tracking server, enabling it to capture the tracked event.

In contrast, other single-event tracking scenarios such as a business insight gathering use case, requires each single-event to be attributed to a specific user, to gather insights over historical data, to create more insightful customer demographics. The service may enrich the behavioural data with a PII or as a generic customer from a specific geographic location (based on IP address). In such scenarios, the *Tracking Pixel* needs the capability to receive a HTTP cookie with a UID or an alternative tracking technique described in previous chapter that can capture a UID. The need to maintain the UID with the tracking event, adds another step of complexity to the anonymous single-event tracking.

6.2.4 Multi-event tracking

Implementing multi-event tracking introduces more complexity to the process, than single-event tracking. It is a collection of related single events that makes up a composite event. For example, CPA advertising model presented in section 2.5.2, which does not pay for *clicks* in visitor traffic generation, instead only for monetary outcomes. One CPA event takes place across three or more different domains. It may happen in one browsing session or over different sessions and at different times. But the composite tracking event, comprises of a click event that happened at an affiliate's website and a conversion event that happened at the e-commerce site. Those web domains who do not exchange information directly between them, but they all communicate directly with tracking domain, where the conversion events are been matched with click events. This adds more complexity, and the tracking techniques that can perform this task are limited, in contrast to techniques available for single-event tracking discussed above. The experiments in this research simulate many categories of tracking scenarios. Researchers and practitioners can choose a technique that suits a given scenario from the techniques presented in chapter 4 and further discussed in chapter 5.

6.2.5 Tracking vectors

Unlike HTTP-cookies, alternative state management methods discussed here are not by design, technologies invented for tracking purposes. Methods that automatically transfer persisted identifiers back to the webserver with each HTTP-request, without having to implement additional functionality,

are good candidates as tracking vectors. It is convenient, reduces the number of points of failure and can reduce latency. By design, both HTTP-cookie and ETags fulfil this condition. Webservers set cookies or the ETags, and on subsequent requests look for the cookies (by the name) or in case of ETags, by the value. As part of the HTTP protocol, it is the responsibility of the browser to return the unique identifier to the server, with every request. In case of “Local Storage” it is not designed to send its values back to the server. It is meant to be used by the code running on client browser. Therefore, additional efforts are required to extract the information from the local storage and post it back to the server.

The “super cookie” concept and associated technologies were not designed to be used for the purpose of tracking; therefore, all alternative tracking vectors can inadvertently become obsolete with new releases of those technologies. As a technology that formed the super cookie concept “Adobe Flash Local shared objects” commonly known as “Flash cookies” have been upgraded by Adobe, to prevent them from being used as tracking vectors. Further, most browsers have by default, disabled access to Flash content and require user’s explicit permission. Ayenson et al. (2011) found that ETag retained their identifier values even when the cookies were blocked in a browser and even when using “Private browsing mode”. Results of this research experiments show that all the browsers now block ETags and Local storage, in both above scenarios. Therefore, keeping abreast with current developments of these technologies will enable researchers to adapt to the changes and modify the techniques to stay ahead of these changing technologies.

However, as seen in the results in Table 15, tracking capabilities using “Local Storage” and “ETags” perform better than HTTP-cookie based traditional tracking technologies, in multiple ways. Most common browsers have a visual indicator on the browser window to show the use of HTTP-cookies within a site. For example, Chrome has a small cookie icon at the end of the URL address bar at the top of the windows. On clicking it, even the least-tech savvy users can delete or even block the cookies to that specific site, thereby failing the tracking process within that browser completely. In contrary,

the use of local storage is not as visible to the user. To view data in the local storage the user must dig deeper, such as use the “Developer Tools” that are accessible to users with more technical sophistication. Nevertheless, deleting HTTP-cookies now deletes local storage too, in newer versions of modern browsers.

It is important to note that the efficacy of tracking vectors cannot be inferred from the number of tests passed, which are listed in Table 15. The experiments were designed to test fail-scenarios, that were discussed in previous literature (Amarasekara & Mathrani, 2016), but each test-scenario does not have the same impact or significance when measuring success related to real-world tracking. For example, the ability to track accurately when using two browser tabs and ability to track despite clearing browser cookies are two different fail-scenarios, which in real-world have different significance to tracking result, and also represent two levels of probabilities of occurrence. Clearing the browser-cache and deleting cookies are two of the most challenging fail-scenarios, that affect HTTP cookie-based tracking systems that are widely in use today. The results in Table 15 show that two alternative tracking vectors, ETags and HTML5 Local storage can both successfully handle one scenario each. Therefore, our proposed *Robust tracking* technique, which uses both above vectors in tandem to supplement the existing cookie-based tracking technique results in an improved tracking capability, which can solve two of the major fail-scenarios.

As ETag values are meant for the caching mechanism and therefore not easily visible to the general user, ETags have an advantage over HTTP cookies and HTML5 Local storage. Also, the tools that are readily accessible on the user interface to remove or block cookies, do not delete the ETags, though they affect both the HTTP-cookies and local storage. But removing browsing data and cache history effectively removes all identifiers including ETags.

During multi-domain tracking scenarios, all tracking code should be executed from within an *iframe*, whose source property should be set to the tracking domain. Though the parent page may be owned

by different domains, the *same-origin* restrictions would not prevent sharing the tracking UID between domains, as all tracking codes have their origin in iframe, which in turn has its source in tracking domain. Both alternative tracking vectors used in multi-domain experiments (ETags and HTML5 Local storage), need to use a JavaScript to access the UID, which must run within the *tracking-iframe*. Implementation details specific to each tracking vector, will be discussed under the specific subsection below.

Though we have displayed how these methods could be used for tracking without using cookies, we do not consider them as alternatives for HTTP-cookies. We recommend using cookies as the primary means for tracking, while using other methods in combination to make the process more robust.

The experiments above also verified the often-unintended information breaches. Following data security breaches and privacy threats were simulated during following technology usage scenarios, which are common in personal and business environments.

Using Local Storage as a tracking vector

In chapter 4 of Artefact Description, in subsection 4.1.2, the physical configuration and technical details of a Local Storage based tracking system used in the test environment was described. In Evaluation chapter 5 under subsection 5.7.1, the implementation details of the test environment were described, and different techniques and processes justified. In this section, further generalisation of the implementation technique is presented, as it can be applied to other use cases and scenarios.

As Local storage is a client-side technology, when used as a tracking vector, the web application needs to take the extra steps needed to retrieve the UID from Local storage and send it to the webserver. At the start of a browsing session, during the first HTTP request, the UID is never available to the webserver. With the first HTTP response a skeleton page should be sent with static content and a tracking-iframe. A JavaScript within the tracking-iframe will then execute on local browser to extract the UID. If personalisation of webpages is desired, the webserver needs to send the UID back to

webserver asynchronously. The personalised content thus generated can then be asynchronously loaded to the browser for display.

If no such personalisation of the page content is required, but the UID needs to be sent to the webserver in response to some user action (e.g., CPC, CPA models), the action hyperlinks such as the *Click Pixel* can be appended with the UID using JavaScript. It can also be sent in a hidden form field if the requirement is simply to send the UID back to the server. When the user clicks the updated URL, the click-tracking server can retrieve the UID from the parameter list of the URL or if posted back as a form field. This tracking process involves multiple steps. A JavaScript running in an *iframe* cannot access the parent page directly. Therefore, we need two JavaScripts; one that is embedded in the parent page, where the banner advertisements (or hidden form fields) are located, and the other embedded within the *iframe*. At the time of loading the page, the JavaScript on parent page registers an *EventListener* that listens to any *messages* from the *iframe*. The JavaScript in *iframe* extracts the UID and posts the message to the parent window, which is then used by the JavaScript in the parent window to update the Click-URLs, as shown in subsection 4.1.2.

During the first visit to any tracked website, a new UID is created, even if the user does not buy a product or interact any further with the website, except viewing the landing page. That UID remains in *Local Storage* without any expiry unlike HTTP cookies, until it is explicitly removed from *Local Storage*. Therefore, it is much more persistent than HTTP cookies.

The tracking results In Chapter 5 (Evaluation) under sub heading *Test 5* shows, that all three tracking vectors failed to track a user under *Private browsing* mode. This result confirms with the specification that requests browsers to treat all storage mechanisms to behave similar to HTTP cookies within a *Private browsing* context. Different browsers implemented this recommendation to a different degree since the introduction of HTML5 Local storage and other web storage mechanisms such as Web SQL Database and Indexed Database. As of now in May 2021, the results show that all tracking vectors

adhere to the recommendation of the specification, by isolating the browser session during *Private browsing mode*. In this mode, browsers do not store the specific browsing session related data, such as the browsing history, content related to the visited sites, passwords used or cookies received (Alabbas & Bell, 2018, 2021; Olejnik, 2019). Adhering to these recommendations have taken time, hence previous research findings over the past decade has been varied. But the intention of this research is to evaluate through live experiments and present the state of tracking vectors as of now, in early May 2021.

Robustness of ETag as a tracking vector

The purpose of the ETag is to reduce network latency and data bandwidth usage, by caching resources that do not change frequently, on user's computer (Fielding & Reschke, 2014). Physical configuration details of ETags as a tracking vector in our test environment was presented in subsection 4.1.3. Implementation details within test scenario was discussed in subsections 5.7.2. In this section, I further generalise some of the implementation details, as it applies to a wider general use as a tracking vector.

By looking at the process 4 in Figure 15, which relates to the tracking-iframe's HTML source being loaded from the tracking server, one would be tempted to use this HTTP request for UID creation and maintenance through ETag. Though the URL is always the same from any Affiliate webpage, it cannot be used for cross-domain tracking, when the URL from the e-commerce page needs to pass parameters containing additional data such as sales amount, Transaction ID, etc., whose values change from e-commerce site to site as well as with each transaction. Similarly, it cannot be used even for single-event multi-domain tracking, even if the URL remains unchanged, because when using ETags, the source of the request, which is reflected by the "Referrer" header, must match too. In the above-mentioned configuration, it can only be used in a single-event & single-domain tracking scenario. Therefore, in all other scenarios, we need to use an HTTP request that can be called from within any affiliate webpage and from any e-commerce payment confirmation page, with the same URL, without any parameters. In such scenario, a viable option is to embed the parameter values into request URL

or into a field content, and attach a JavaScript file, which can then make an asynchronous call to the tracking server's UID returning URL. Next, the JavaScript can combine the UID with the parameters passed by e-commerce server and make another asynchronous call to the conversion tracking URL.

The ETag specification (Fielding & Reschke, 2014) requires the webserver to return a 304 (Not Modified) response, if the ETag sent by the client browser matches with the server version of the resource. But, when the ETag usage is repurposed as a tracking vector, a 2xx result code should be returned, instead. Returning a 304 (Not Modified) result code, halts any further processing of the server response, by the browser. It is also important to set a new ETag header with each server response, using the same UID. As per the current version of ETag specification, the default behaviour is: if the server returns a 304 result, the browser continues to retain the ETag for future use. If a new ETag is returned instead, the browser will replace the old ETag with the new. If no ETag is returned with the 2xx result code, then the ETag would be removed from the resource, which would result in losing the UID for future use and causing a tracking failure. A future update of the specification could change this behaviour to abandon the ETag, if a server returns a response with an ETag that is identical to the ETag it received, together with a result code that is not 304 (Not Modified), as that would indicate tracking behaviour.

Versatility of Robust tracking

A chosen set of tracking vectors was evaluated within different use cases, as described in the two previous chapters. Experiment results under different *fail-scenarios*, shown in Table 15 indicate that Local storage and ETag based tracking techniques can each track successfully under two of the most critical fail-scenarios, i.e., when the cookie collection of a browser is cleared and when the browser cache is cleared of its contents. As a result, we combined the use of *HTTP cookies* with *Local storage* and *ETags* resulting in a more reliable tracking technique and named it as *Robust tracking* method.

When using *Robust tracking*, the UID is saved within the storage mechanism of each tracking vector, so that if one tracking vector fails, the next tracking vector can be used to retrieve the UID. One of the

several different algorithms can be used for this, depending on the requirement. The least work-intensive might be to check each UID storage, based on ease-of-access: i.e., first check the local storage, or HTTP cookie which happens at client-side, and if found, update the click URL with it. If none found or if the cookie is set with server access only, then a round-trip to the server is required to check the cookie and ETag. This approach is based on finding the first available UID occurrence. I have chosen a different approach, that provides more robustness, as shown in Figure 16; it entails additional steps to synchronise all UID storages with each tracking instance, thus automatically correcting any previous tracking malfunctions, if any occur. In a rare chance, should one of the UIDs gets overwritten with a new UID, it will be corrected with the next tracking event. In such situation, the UID's will be compared, and the oldest UID will be restored across all UID stores.

Respawning

Cookie respawning was discussed and demonstrated by Ayenson et al. (2011), which is essentially to recreate deleted tracking cookies. They used Adobe's *Flash cookies*, which were indestructible during the time of their research, to back-up the UID. If a user clears the browser cache and deletes existing cookies, the web application was capable of restoring the HTTP cookie with the identifier found in the *Flash Cookie*. Though Flash cookies are non-functional now, we have demonstrated here, how to respawn the HTTP Cookies using HTML5 Local storage and ETags. Processes 6 to 11 on the *Robust-clicking* sequence diagram in Figure 16, show the respawning process at work. Table 15 shows that ETags are very resilient tracking vectors that persist against cookie deletions. Therefore, processes 6 to 11 demonstrate, how the UID within all three tracking vectors stay synchronised during each tracking instance. If any disparity in UIDs is found between the three tracking vectors, either due to an application error, wilful manipulation or due to any such uncommon event, the respawning process queries the UID table in the database server and chooses the UID that was issued first out of the two unmatching UIDs and respawns that earliest issued UID among all three tracking vectors. As HTTP cookies are most vulnerable and most accessible to users, causing them to be deleted, the respawning process increases the robustness of the tracking capability considerably.

Newer versions of web technology specifications keep evolving responding to the threats and user-privacy concerns. At the same time, practitioners and researchers find newer ways to forego those restrictions and adapt to those changes. For example, since browsers started implementing session isolation during *Private browsing* mode, current trend in tracking algorithms involve first verifying if a user is in *Private browsing* mode, when a new user is detected. If Private mode is detected, it will attempt respawn the UID through an alternative tracking method. There are several different ways to identify if a user is using *Private browsing* mode, and more techniques are found regularly. An empty browsing history, or checking the CSS for visited links, and errors popping-up when trying to save data to different local storages (as saving locally is disabled in private browsing), were some of the giveaway signs of a user in *Private browsing* mode. The ability for the practitioners to identify a user in *Private browsing* mode, negates the advantages of the privacy mode to some extent. In response to this cat and mouse game, latest specifications recommend browser manufacturers to counter such detection methods by, for example, requesting browsers to make sure that error conditions do not pop-up when trying to save data to local storages, instead offer identical functionality as in normal browsing mode; then to discard the data at the end of the browsing session, thus preventing tracking technology developers from discovering a user in *Private browsing* mode in the first place (Olejnik, 2019).

From an E-commerce perspective, the inability to identify a user with a previously assigned UID during *Private browsing* mode does not have a major implication. Many e-commerce transactions and traffic generation models (i.e., CPM, CPC, CPA) do not require the ability to identify a user with PII or over a longer period, instead tracking capability from the beginning to the end of a transaction, usually suffice. Multi-event interactions such as clicking a banner advertisement at an affiliate website and purchasing a product at an E-commerce site happens frequently within the same browsing session. In such cases, session cookies and other storage mechanisms that are used in *Private browsing* modes do function as much as they do under normal browsing modes. The click- and conversion-tracking as well as commission payments to affiliates will continue to function with session storage capabilities.

Users may intentionally switch to *Private browsing* mode, usually when they visit a website, where they expect extra level of privacy, not to sabotage an affiliate earning a commission as a payment for Internet traffic generation. It can mostly effect tracking service providers who track users to gather behavioural information to generate electronic personas or to provide business analytics, a process which need to gather such information over longer periods than within a single browsing session. That is exactly the kind of information a user does not want to allow third parties to gather, hence the merits of privacy vs. tracking provider needs, are beyond the scope of this research.

6.3 Research goal 3

The third and final research goal is to verify through experiments and describe a tracking privacy model based on levels of privacy intrusion, during diverse tracking activities. Through “robustness improving experiments” (artefacts described under 4.1 and results discussed under 5.4) I have demonstrated that most e-commerce activities and web traffic generation activities such as AM can be carried out in a privacy-preserving manner. Those experiments used generic UIDs without any PII attached to them. This provides a pathway for e-commerce operators who are committed to privacy preserving tracking practices, an improved robust and privacy-preserving tracking technique.

Nevertheless, some service providers may choose to adopt the techniques to improve the robustness, presented in this research, but may not adopt the privacy-preserving tracking techniques that are presented if their business models are based on gathering customer demographics and creating marketable digital personas. Hence, this research proposes a privacy model, based on *Information Seeking Behaviour* (ISB) of different business model use cases and their reach, which can aid regulatory frameworks to control user-privacy. The privacy related tracking artefacts were described in subsection 4.2 in *Artefact Description* chapter and the results of the experiments were evaluated and described in subsection 5.5 in *Evaluation* chapter. In this subsection, I generalise the findings, as it applies to online privacy intrusions within a wider context.

6.3.1 Information seeking behaviour

As we investigated privacy intrusion levels that occur during different tracking scenarios, a clear distinction emerged that set the different tracking use cases apart. The differences were based on privacy intrusion levels, which has a positive correlation to the ISB of the tracking application. *ISB Level* was defined as a variable of the *user-privacy* construct that would be measured during the experiments to determine the privacy intrusion level of different tracking use cases. ISB of a tracking application is intentional in most use cases which are described below, but it is an unintentional consequence in case of *tracking as a pure technical necessity* use-case. The ISB is the causation of privacy intrusion; parameters of privacy intrusion levels are limited by the technological limitations within each use-case that we will discuss under each subsection below.

Under the research goal 3, a main objective is to establish a clear definition for *Tracking as a pure technical necessity in cross-domain interactions* in a stateless ecosystem such as Internet. This will enable privacy legislators and policy makers to recognize the technological requirement boundaries in future endeavours, thus not hinder the advancement of technology. At the same time, it enables them to curtail predatory information seeking behaviour of specific actors and to safeguard the privacy of users. This will also facilitate, for the users to make informed decisions when choosing privacy preserving settings within their daily online environments.

As users, privacy groups and countries are getting more concerned about privacy and user rights, more regional and local regulations such as GDPR (GDPR, 2016) are being implemented that restrict online tracking activities. Many of these legislations fall far short of intended goals (Matte et al., 2020; Papadogiannakis et al., 2021; Utz et al., 2019); at the same time, they can have a detrimental effect on non-privacy invasive tracking techniques that are an essential technological necessity in a stateless ecosystem. Therefore, a detailed and systemic approach to ISB and the resulting privacy intrusion through a privacy model will help the legislators and researcher alike, when targeting specific groups and techniques in future endeavours.

Tracking activities can be divided into five categories based on ISB and resulting privacy intrusion levels. Each category of tracking has different levels of privacy implications therefore should be addressed differently:

(1) Purely technical: Tracking process used in AM and different web traffic generation models fall into this category. E-marketing methods necessitate the ability to track a visitor from the web traffic source until completion of the transaction. ISB is limited to a non-PII UID assigned to each user during first visit. A “click-Pixel” in all advertising pages (e.g., banner advertising) and one “conversion-Pixel” in each payment-confirmation page are the only tracking requirement for this kind of tracking technique. This mode of tracking does not create privacy concerns to the users, therefore new regulations and policies need to consider the importance of current and future technological necessities during policy and legislature planning. Implementation details are presented in chapter 4 under *Artefact Description* and implementation evaluation in subsection 5.5.1.

(2) Non-PII based: A website or an e-commerce site that tracks visitors to its own domain, to gather basic customer demographics, fall into this group. There is a clear ISB in the process, but the information is limited to what is sent by the client’s browser to the webserver, with each HTTP request. A visitor can be “remembered” and only visits to the same local website can be tracked over a long period, using a UID which does not reveal any personal information.

The IP address within each session can identify the current location of the visitor, by the country and often by city in larger countries. This can therefore reveal a visitor’s travelling behaviour, and the language can be inferred from the IP address location or from the browser’s language setting. A combination of both is a more reliable option, as a person travelling outside of home country, but using own device, can be accurately identified with the browser language, though language of the geo-location may differ. On the other hand, a travelling visitor using a public computer in a net-café would have a different browser language to her own. The operating system (windows, Linux, or apple)

can be useful for profiling to some extent, and the mobile device type and brand, in case of mobile users, can add to the demographic information.

Though such websites do not usually require users to be logged on to browse through pages, e-commerce sites do require a user log-on, to purchase goods. Therefore, such e-commerce sites usually gather some PII data such as Name, delivery address, telephone, and email data. Any e-commerce site that combines anonymous user profiles discussed above with the PII-based local login account, has a limited ISB.

Experiments carried out are evaluated in subsection 5.5.2. It should be noted though, that the data relating to a customer's web interactions are limited to the visited domain and personal data is limited to the information provided by the user during account creation. A user has the option of not providing non-essential information such as OSN profile URLs, business affiliation details, etc.

(3) Non-PII based external provider: The tracking process used by AM networks and tracking service providers, who venture a little further than being purely a tracking service provider, fall into this category. Such services provide business analytic services as a premium product, displaying a strong ISB, but without using PII. They can provide business insights beyond one's own domain boundaries, but their reach is still limited to the network of e-commerce sites that have subscribed to the service provider. They can create a useful set of data regarding product interests and purchase habits that was created using the behavioural information gathered across many e-commerce sites, including one's business competitors. Any products that were perused at other domains give away the current purchase interests of the visitor. The knowledge of non-purchase of a product at one site, can be sold to the next site as a premium lead such as the "remarketing" leads provided by many such services.

At this level of tracking, despite knowing the approximate location, language and buying habits, the visitor is only identified by a UID, without any PII. Visitors to a website do not see any presence of a

third-party tracking provider, nor require any login to a tracking service; hence no PII is captured, and tracking takes place in the background without visitor's knowledge or explicit consent. When an e-commerce site has subscribed to an external tracking provider, a *Tracking Pixel* embedded in webpages of the e-marketing site, causes a connection to the third-party tracking service, which allows the tracking service to capture information about the visitor. This process is described in chapters *Artefact Description* and *Evaluation*, in detail. One of the major differences between this category and PII-based OSNs category discussed next, is that the service providers at this level do not have a product or service that has a global reach.

(4) PII-based OSNs: This category can span from harmless and non-privacy intrusive services to information-scavenging nefarious operators. At the lower end of the scale are the e-commerce and other service providers who may choose to gather more PII than what a local login account can provide, hence use an OSN that provides external authentication, such as Facebook, Twitter, etc. At the opposite end of the scale are OSNs with a global presence, such as Facebook, Twitter, LinkedIn, etc., who have large amounts of personal data, most of which has been provided by users through social media posts and photo sharing applications. In between, are entities gathering business intelligence who are mostly invisible to the users, which can include entities such as Cambridge Analytica (Berghel, 2018; Manokha, 2018; Margaret, 2020; Richterich, 2018; ur Rehman, 2019). It also includes data brokers such as *Axiom*, *Experian* or *PeekYou* who gather data from different OSNs and combines them with data available at other sources (census, voter registrations, court reports, driving records, etc.) to create rich datasets that are sold as a commodity to interested parties (Manokha, 2018). Actors in this area of operation either do not have any visible web presence (e.g., Cambridge Analytica) or appear as a free service (e.g., Facebook, LinkedIn), sometimes very useful to the users. Web scraping and web crawling activities form an important part of their activities. They usually offer free services and tools, requesting subscribers to by fill out forms with PII such as names, contact e-mails, etc., so that they can place tracking cookies into the browsers of unsuspecting visitors.

Providers of categories 4 and 5 are set apart from the other categories due to one or more of their products or services having a global reach. The global reach provides them the crucial advantage that millions of customers around the globe will have an account with them. Different service providers who sit at the top of the tracking hierarchy (Figure 28) have different methods to track the users across the globe. OSNs like Facebook, Twitter, LinkedIn can gather visitor-data at partner sites, either through *oAuth* external login service they provide at partner sites or using “Like” and “Share” buttons of different services. “*Off-Facebook Activity*” is another way that businesses and organisations share information with Facebook, when partner sites use *Facebook Pixel*, *Facebook SDK* or *Facebook Login* feature. Facebook has now provided a new tool enabling users to view a list of organisations that share *Off-Facebook* information in their *settings* page of the profile, shown in Figure 25, which requires a user to be logged-in to the account. Facebook uses this information to personalise the advertisements they show on Facebook, based on the user’s recent activity, outside of Facebook (Facebook, 2021a). The explanation of the process described by Facebook in Figure 25 is downplaying the gravity of the privacy intrusion.



Figure 25: How "Off-Facebook activity" is sourced (Facebook, 2021b)

This example shows that tracking activities of some actors fall within one of the above ISB categories, while others such as Facebook can have different business models that span multiple ISB categories. In this example, the hierarchical nature of tracking ecosystem depicted by the reversed pyramid (Figure 28) allows operators in higher levels to have oversight and capabilities of all categories below it.

Most websites today, implement a Business analytic service such as Google's *Universal Analytics*. A *Pixel* embedded in the page triggers a post-back to Google analytics servers that can identify the user by Google identifier. That enables Google to capture all online activities of a user, such as the sites visited, specific actions such as online purchases, which are accumulated over a long time. This is a top-down tracking approach for user profile creation, where first the person is positively identified, and then over time, behavioural data is accumulated to enrich the digital persona. Microsoft, Apple, Google, Facebook, Twitter, LinkedIn, and other OSN companies fall into these two categories 4 and 5 (Krishnamurthy & Wills, 2009; Mayer & Mitchell, 2012). Most people have an account with one or more services of these tech giants.

This is the kind of tracking activity that is most frowned upon by users and most privacy laws are trying to curb. The comprehensiveness of the created user profiles depends on the visibility a service provider has over the Internet. Google trackers are found across 80% of world's leading (Alexa top 1 million) websites (Cahn et al., 2016; Libert, 2015). Online tracking by the largest third-party organisations has grown from a 10% in 2005 to 20-60% by 2008 (Krishnamurthy & Wills, 2009).

OSNs of this category and browser manufacturers of the next category, both have the highest level of ISB, among different categories presented here. Their business models are based on marketing comprehensive digital personas as a commodity to other OSNs, Enterprises, security agencies, governments, and political parties (Hinds et al., 2020; Manokha, 2018; ur Rehman, 2019). Groups 4

and 5 have similar ISB and modus operandi, but they differ based on the tracking limitations they face within a browser environment.

(5) PII based Browsers, Browser-extension, and OS manufacturers: Very similar to the above category 4, with similar ISB, but who have the capability to circumvent the privacy-related restrictions imposed by the browsers on websites. Browser extensions are small software modules, that can provide customised behaviour to browsers. They can be installed by the users on their browsers. Extensions that claim to block advertisement or manage cookies, expose tracking, etc. are popular among some users. Operating system manufacturers Microsoft, Apple and Google have the advantage of operating system level identity knowledge (Gamba et al., 2020).

Most privacy preserving features such as *Private browsing* mode, Cross-site scripting, CORS, etc. are client-side browser implementations. For example, a webserver still sends the HTTP cookies even across domain boundaries as discussed in 2.3.1 and 4.1 to the browser, and the browser will determine, based on CORS restriction headers, if the received HTTP cookies should be stored in cookie-cache or discarded.

Most popular browsers have a log-in feature, though browsers can be used in an anonymous mode without log-in. Logging in provides an improved user experience, by remembering passwords, browsing history, form data for frequently filled form fields and perhaps most importantly, synchronising data and sessions across all the devices used by the user. For users who don't use Chrome browser, Google offers *Google toolbar*, which, in effect, allows similar visibility. While it provides convenience to users, it also provides the tracking operators complete oversight on user interactions on Internet and the ability to create comprehensive personas.

In a *Connected World* today, people use multiple devices: wearables, mobile phones, tablets, work PC, personal laptops etc. It is convenient to move between devices, continuing tasks started on a different device. That provides uninterrupted connectivity to the user, and continuous tracking capability to the

category 5 service providers. For example, people who use Chrome browser by Google, log in to the chrome browser usually on first use. In turn, Google has a record of user’s activity throughout the Internet, not only at websites owned by Google. The GPS on the user’s phone provide the physical location at any given time. A quick look at my *Timeline* on *Google Maps* shows me that I have activated the *Timeline* feature on 2nd December 2015. A detailed movement history is available to Google since ca. five and half years, to date. The Figure 26 shows start time, walked distance to and from with exact route on a map.

Google Maps Timeline feature is turned off by default. But Google support documentation mentions that even if *Location history* is not enabled by a user, the location data can continue to be saved by other Google services (Google, 2021a).

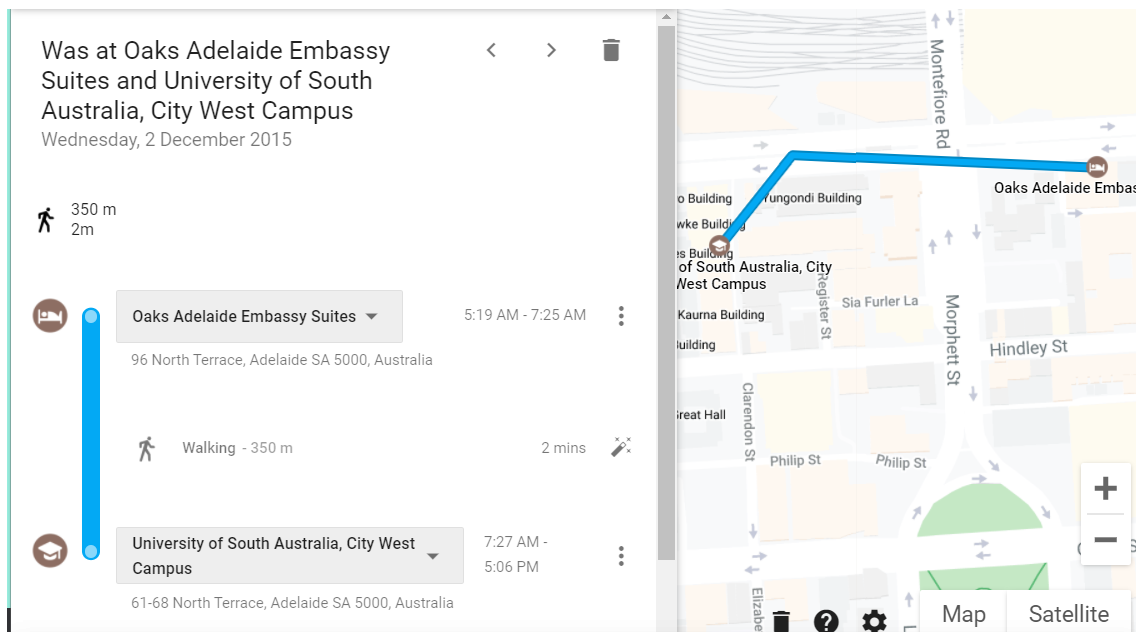


Figure 26: Google Timeline - Location history

The *Google Maps Timeline* can summarize a monthly report (Figure 27) that presents some of the information aggregated by Google services. It may be convenient and informative to an interested user but reveals how daily movement information and visit of places including shops and other commercial venues can provide comprehensive marketable insights to a person’s life. While other

researchers have examined how the information is used by Google, Facebook, Cambridge Analytica, etc., it is beyond the scope of this research (Hand, 2018; Manokha, 2018; Persily, 2017; Richterich, 2018; ur Rehman, 2019). Instead, based on data science, possible use-case capabilities can be inferred from data summaries presented by the services.

Ravi, here's your new Timeline update

You're receiving this monthly email because you turned on Location History, a Google Account-level setting that saves where you go in your private Timeline.

Location History data also helps give you personalized information on Google, including better restaurant recommendations, and suggestions for a faster commute. You can view, edit, and delete this data anytime in Timeline.

[Explore Timeline](#)

Location History: ON
[Manage Settings](#)



Your April visits

8

Places

3 new



Your April activity



47 km
2 hr

Highlights Places visited



Lucky Fortune Restaurant (喜运来酒家)
New



Herb Morgan's Tyres & Wheels
New



Avondale Jockey Club

[See all visited places](#)



Your all time data

8
Countries/
Regions

38
Cities

234
Places

Manage your Location History

Visit your private Timeline to view, edit and delete your Location History

[Go to your Timeline](#)

Pause your Location History

Visit Activity controls anytime to pause your Location History

[Go to Activity controls](#)

Figure 27: Google Maps Timeline

6.3.2 Tracking privacy model

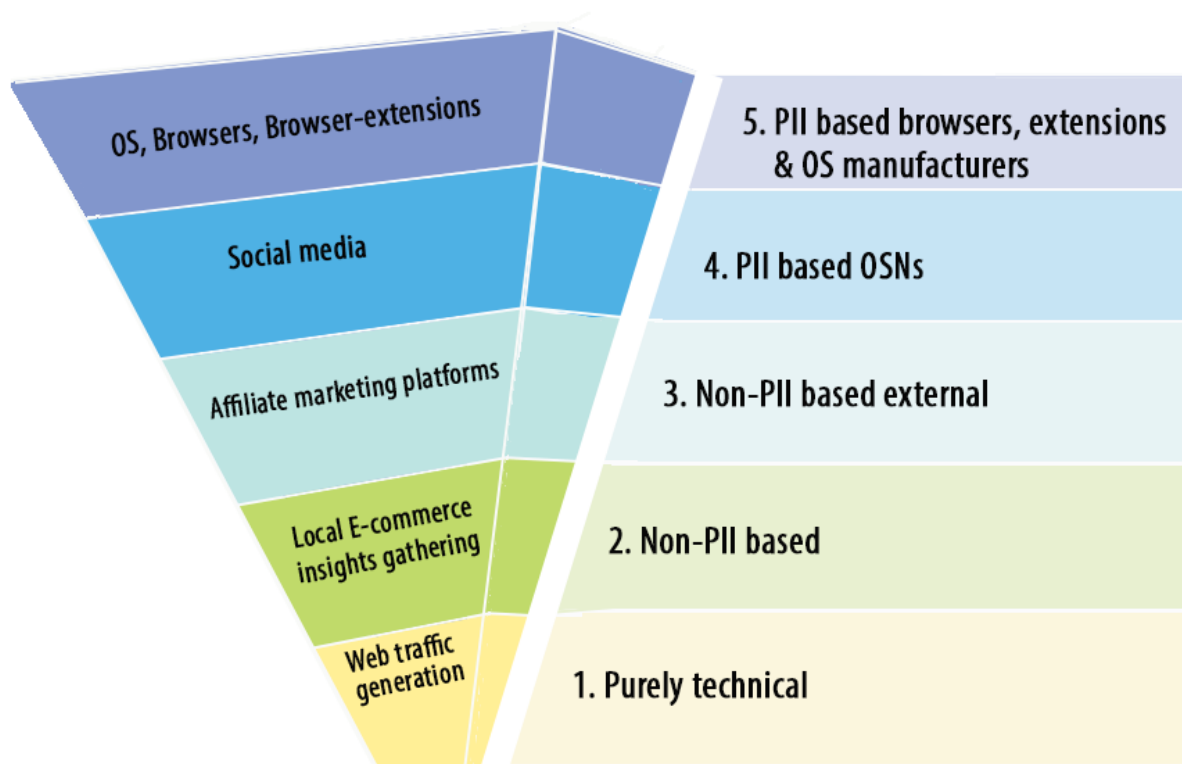


Figure 28: Tracking privacy model relating to ISB

Privacy intrusion level shows a positive correlation with ISB of the tracking application, as discussed in previous subsection 6.3.1. Figure 28 shows the hierarchical nature of ISB of different tracking use cases over privacy intrusion level. It represents the five categories of ISBs described above, where lower category numbers at the bottom of the figure represent lower ISB which result in lower privacy intrusion. An inverted pyramid was used to show the influence over the cyber-world, by the mass of the layer at each level. It shows the operators at lower levels have a limited scope and visibility of user-activity, such as only within their own domain. Moving up the inverted pyramid, players above have not only more intrusive tracking techniques, but also wider visibility of user activity. Actors at the very top, have visibility over the whole Internet. The privacy model can also be represented with a standard pyramid representing the dominance instead of the reach. In that case the mass of the pyramid at each level representing the number of operators and the level representing the

dominance. The apex representing Level 5 consists of a few browser manufacturers, but with a dominance and visibility over the entire cyber-world, and the ability to track a user across the Internet.

An operator can function at multiple ISB levels simultaneously, sometimes unknowingly. For example, a website that has subscribed to a non-PII based external service provider at level 3 might also implement an external login service provided by Facebook or incorporate a “Like” or “Share” button, or it might have used Google *Tag Manager* or *Universal Analytics* to receive business analytics. That operator functions as a non-PII based ISB at Level 3 for itself, but unknowingly functions as a Level 4 PII contributor for Facebook or Google. The level 4 and 5 providers will collate the data thus received, with their already existing huge repository of user profile data to enrich the existing digital personas, to create a highly marketable commodity.

The *Cambridge Analytica* scandal has revealed that there are other actors at level 4, who do not have any visible interface nor provide any useful service to the users, or other level 4 and 5 operators, who scrape data off the level 4 repositories. Such undercover operators market digital persona-based services selectively to the largest premium customers such as governments, political parties, large conglomerates to influence public opinion (Afriat et al., 2021; Bakir, 2020; Laterza, 2018; Manokha, 2018). That exposes even more vulnerable situations like “data thieves” (Facebook) becoming victims of other data thieves (Cambridge Analytica), harvesting tens of millions Facebook profiles (Bakir, 2020).

An example of a digital persona that can be created at each ISB Level is presented below (Table 17). It is intended to provide what kind of data, and which level of privacy exposure that can take place at each level.

Table 17: Digital personas at each ISB level

Category 1. Purely technical
Digital persona
User "CX16370300517261265490" clicked ecotourism.com banner at nztravelguide.net on 01/01/2020. Monetary outcome of 50.00NZD bearing TransactionID 12345 occurred on 05/4/2020.
Category 2. Non-PII based
Digital persona
User "CX16370300517261265490" / Jane DOE of 30 Arran Street, Auckland, email Jane@freemail.com clicked ecotourism.com banner at nztravelguide.net on 01/01/2020. Monetary outcome of 50.00NZD bearing TransactionID 12345 occurred on 05/4/2020. As per IP address, from New Zealand, speaks French (browser language) uses an iPhone, has visited the site 15 times since first visit 10/5/2018.
Category 3. Non-PII based external
Digital Persona
Jane DOE of 30 Arran Street, Auckland, email Jane@freemail.com. As per IP address, from New Zealand, speaks French (browser language) uses an iPhone, has visited 520 affiliate sites 1500 times since first visit 10/5/2018.clicked ecotourism.com banner at nztravelguide.net on 01/01/2020. Had following interactions: Purchased ->Flight CMB -> 500.00NZD -> a.com -> TransID 12345 ->on 5/4/2020 ->3 visits Perused -> Hotel room CMB -> ->b.com -> -> on 5/4/2020 ->2 visits Perused -> Hotel room CMB -> -> c.com -> ->on 6/4/2020 -> 1 visit Purchased -> Travel Insurance ->300.00NZD -> d.com ->TransID 43435 ->on 6/4/2020 -> 1 visit
Category 4. PII based OSNs
Digital persona
Jane DOE, Female, 36 years, of 30 Arran Street, Auckland, email JaneDoe@freemail.com. As per IP address, from New Zealand, speaks French (browser language) uses an iPhone. She works at: (from LinkedIn profile). e-marketing specialist -> e-solutions.com -> 2010-current

event manager ->Event Management Ltd. -> 2007 -2010

Education:

Postgraduate Marketing, Bachelor of Business studies

Political views: centre-left

Responds happy: animals, cats, cooking, puzzles,

Responds sad: animal cruelty, destruction of nature, racial discrimination, gun control, SOS

Responds Angry: MAGA, animal cruelty, ecological negativity, Nationalism, capitalism,

Shared posts for: Greenpeace, animal, cat, animal rescue, anti-palm oil, ethnic

Liked: Democrats, Labour, herbal, vegan, vegetarian, travel

Category 5. PII based browsers, browser extensions and OS manufacturers

Digital persona

Jane DOE, Female, 26 years, of 30 Arran Street, Auckland, email Jane@freemail.com. As per IP address, from New Zealand, speaks French (browser language) uses an iPhone. She works at: (from LinkedIn profile).

e-marketing specialist -> e-solutions.com -> 2010-current

event manager ->Event Management Ltd. -> 2007 -2010

Marketing Assistant -> e-solutions.com ->2006->2007

Education:

Postgraduate Marketing, Bachelor of Business studies

Political views: centre-left

Responds happy: animals, cats, cooking, puzzles,

Responds sad: animal cruelty, destruction of nature, racial discrimination, gun control, SOS

Responds Angry: MAGA, animal cruelty, ecological negativity, Nationalism, capitalism,

Shared posts for: Greenpeace, animal, cat, animal rescue, anti-palm oil, ethnic

Liked: Democrats, Labour, herbal, vegan, vegetarian, travel

Had following interactions:

Purchased ->Flight CMB -> 500.00NZD -> a.com -> TransID 12345 ->on 5/4/2020 ->3 visits

Perused -> Hotel room CMB -> ->b.com -> -> on 5/4/2020 ->2 visits

Perused -> Hotel room CMB -> -> c.com -> ->on 6/4/2020 -> 1 visit

Purchased -> Travel Insurance ->300.00NZD -> d.com ->TransID 43435 ->on 6/4/2020 -> 1 visit

There have been some research, that investigate different use-case scenarios of such digital personas by Level 4 and 5 actors. Bakir (2020) concluded that psychographic profiling and targeting carried out by Cambridge Analytica is a form of psychological operation (“psy-ops”). Bilge et al. (2009) demonstrated profile cloning (creating cloned profiles of an existing user on an OSN) and cross-site profile cloning (creating a profile of a user who has an existing profile on one OSN and impersonating that user on another OSN) exploits undertaken by third-parties without the consent or knowledge of the OSNs involved. Research study of Hu et al. (2007) involved predicting user demographics based on user’s browsing behaviour. Level 4 and 5 actors can enrich the digital personas they have created by combining a steady stream of demographic and psychographic information based on activities of a user across large number of web domains and use this information to influence users, their choices, opinions, affiliations with groups such as political parties, religious or environmental groups and social movements etc., which have been discussed in previous studies. It does not come as a surprise, due to the large amount of data gathered by Level 4 and 5 operators. The information can be distilled in to features that are important to the provider, that model different aspects of user-behaviour. Machine learning techniques can further enhance the correlation of these features with expected influencing outcomes, similar to risk prediction based on a user’s browsing behaviour carried out by Canali et al. (2014).

It is equally interesting to see, how operators such as Cambridge Analytica, operating in a stealth manner can also tap into the vast reservoirs of information that belong to Level 4 operators, such as Facebook. It came to light in 2018 that, together with Cambridge researcher Aleksandr Kogan, who developed a personality and quiz application called “*thisisyourdigitallife*”, Cambridge Analytica was able to harvest millions of Facebook user profile data, without the knowledge of Facebook, as claimed by Facebook (Bakir, 2020; Brown, 2020; Laterza, 2018; Manokha, 2018; Richterich, 2018; ur Rehman, 2019).

6.3.3 Information scavenging

Even without such software artefacts used by Cambridge Analytica, any third-party operator can harvest personal information from OSNs to create elaborate digital personas. Experiments to demonstrate the exploits and the methods described here were not tested during this research in this exact manner, as obtaining ethics approval for research relating to human subjects and the experiments were beyond the scope of this research. Instead, the technological feasibility in practice was experimented, within the AMNSTE22 network. Social media posts were simulated through multiple pages categorised as those representing diverse interests mentioned below, and visits to those URLs were captured at the tracking server. Queries run on the captured data simulated the information harvesting and digital persona creation techniques which together with empirical knowledge enables to present the following techniques to harvest user information from OSNs.

An operator who wants to gather customer demographics and combine them with psychographic information can start by determining personality traits expected to capture. OSN user profiles with names that reflect the main personality traits can then be created, as the profile name should have some relevance to the posts. Using each profile, posts that invoke emotional responses in either extreme (extremely positive/negative and in-between for finer gradience of evaluation) will then be made, and user actions such as *Likes, Shares, Sad, Angry, Surprise* etc. can be classified for each personality trait from each responding user. For example, under political views, a post depicting

extreme right-wing and extreme-left wing and those of the centre, can determine the political affinity of the responders. Repeating similar posts can increase the accuracy. *Share* action can also reveal the “circles” as used by Facebook. Closeness of relationships between users is defined by “circles”.

Information scavenging act can be taken a step further by creating a *Current affairs* website similar to many online news channels. Freely available content management systems (CMS) can create a very professional looking site within minutes. Similar to posts on OSNs in previous example, articles of diverse interests can be created under current affairs, politics, nature, Home & Living, etc. and as mentioned above, with articles that are controversial and bordering both ends of the extremes. The URLs of these articles are then posted on a social media. For example, Facebook provides a small preview with URL for external links, and users who click on the link to read the article arrives at the news website of the tracking operator. That allows the tracking operator not only to capture the information of users who responded to that specific topic, but also store a tracking vector such as a cookie in client browser. That enables tracking over a long period, gathering psychographic information on how the user reacts to opinionated views. This knowledge enables targeted opinion-swing-campaigns for each group more effectively.

It is evident that there are such operators using similar tactics, already profiling users. By presenting some of the techniques used, it is hoped that regulatory authorities and industry partners will be able to formulate measures and legislations that can target specific practices, while not curtailing tracking technologies, in general.

6.3.4 Tracking data spillage

There are instances when actors who gather user-related data, inadvertently expose the gathered data to external parties, thus creating additional privacy breaches than what is visible. As discussed above, while Facebook was gathering large quantities of user data, Cambridge Analytica was harvesting Facebook’s gathered data, for many years, without Facebook’s knowledge. Further,

research experiments that simulated real-world AM network using AMNSTE2 exposed some data leakages that happen within AM environments.

The range of information exposed to third-party tracking service providers was observed. E-commerce practitioners who subscribe to services of an AMN expect the AMN to monitor only transactions belonging to affiliate-generated web traffic. Instead, as the tracking *Pixel* is placed on the payment confirmation page and a confirmation page is sent to every customer at the end of a payment, it triggered the conversion tracking process for every transaction. This results in all transactions being captured by the tracking server, including those visitors who came through organic searches, paid advertising, search-engine advertising, and every other traffic generation method. The tracking server can easily differentiate the AM generated traffic from non-AM traffic by the presence of an accompanying HTTP-cookie, which has been placed by the tracking server during click-tracking process. When tracking results are reported, all web traffic that does not have a tracking cookie are excluded by the tracking server and those are classified as non-AM generated traffic. However, enterprises are usually unaware that the tracking service has captured all online sales data of the subscribed e-commerce practitioners.

This information leakage becomes more critical with the popular practice of using services such as “Google Tag Manager”, where e-commerce sites link their *Pixel*-code via the Tag Manager URL instead of triggering directly on the tracking server. This exposes all online sales data to two different service providers (Google and Tracking service), both of whom could use that information to generate additional value-added services, that are useful for the marketing efforts of competitors. For instance, remarketing sales leads that are offered at a higher price are based on the information on unsuccessful sales at competitors’ e-commerce sites, since tracking service providers have visibility over customer interactions within all sites that have subscribed to their services. Some business managers who are uninformed about the information security breaches that occur, and the associated disadvantages choose to ignore security risk over the convenience of analytics (when their sales data are combined

with the rest of business analytics data). Google's Universal Analytics guidelines make end-user privacy policy explicitly a practitioner's responsibility. Their terms and conditions state: "When you implement Universal Analytics, it is your responsibility to ensure that your use is legally compliant, including with any local or regional requirements for specific notification to users" (Google, 2021b).

Google's analytics service which is offered free of charge to businesses, offer complete visibility of user activity that are of commercial interest. *Google Tag Manager* provides Google with visibility over transactions that are considered confidential between an advertiser and AMP within an AM scenario.

6.4 Privacy and perceptions

The approach of this privacy model is based on the ISB of applications and platforms that provide services to users. It differs somewhat from a model that is based on an individual user's privacy perspective. This model is intended to provide the context for regulatory authorities to formulate legal frameworks that aim at curbing certain practices while not hindering technological advances for a connected world. It is also intended to provide application developers, web security implementers and researchers to implement their solutions to align with different levels of the model.

Operators at Level of privacy model freely admit in their documentations that users and their activities are being tracked for advertisement personalisation as shown, for example in Figure 25. It is popular belief that tracking of even very personal data is carried out by tracking algorithm driven automated processes, not by humans; as a result, user's personalised digital environment is a benefit to user, as often claimed by these operators. As behavioural research discussed later in the chapter show, most people are not too concerned nor bothered to act against such intrusions.

The most nefarious practice related to tracking activity today, is the least discussed *opinion swaying* business model. Undertaken by few large Level 4 players, catering directly to large customer entities, such as governments, political parties, and industries, it is away from public eye and easily dismissed as conspiracies. The Cambridge Analytica scandal exposed the tip of the iceberg which gave rise to the

few scientific research undertaken on this topic (Afriat et al., 2021; Bakir, 2020; Berghel, 2018; Laterza, 2018; Manokha, 2018; ur Rehman, 2019). Tracking for advertisement personalisation intrudes only the privacy of individuals, but opinion swaying use-case has far more dangerous and far-reaching consequences that effect whole societies and countries. This will be the major market that is poised to grow hugely in the coming decade, which need to be targeted through privacy regulations. The political will may remain questionable.

What amounts to privacy, what are the accepted levels of privacy exposure, individuals right vs. collective responsibility, etc. are research questions for a social science research, which I do not attempt to answer in this research. The answers can vary based on country, cultural values, etc. Even before the Cambridge Analytica scandal got publicised, Afriat et al. (2021) found in their research work that most participants perceived privacy as a commodity that can be traded, rather than an integral part of one's civil rights. Another research by the authors after the Cambridge Analytica scandal showed the users considered that it is Facebook and other OSN's right to profit from activities on their platforms and it is the responsibility of the user to manage their own privacy. This contrasts with the expectation of privacy campaigners (Boyd & Hargittai, 2010), that such an event would enlighten the public about precarious privacy preservation, currently in practice.

Most traditional advertising models we experience in our daily lives are not customised for our individual preferences. It remains true, that online advertising will drive our online user experiences and under those circumstances, the advertisements that we are obliged to deal with being based on our interests is a consolation than random unrelated advertisements appearing during all our online activities. Personalisation of advertisements require the capability to track our recent activities, which can be accomplished by non-invasive tracking scenarios taking place in Level 3 of the above privacy model. More comprehensive personalisation is carried out using personality traits tracking at Level 4 and 5. While many privacy-conscious users and organizations may consider any level of tracking to be

a breach of individual rights to privacy, others may find them, from tolerable to useful under circumstances.

Chapter 7. Conclusions & future direction

Online tracking is a technological necessity in a connected world, driven by e-commerce activities. Nevertheless, online tracking is associated with privacy concerns. Some research studies are primarily concerned with privacy protection, which can hinder the user-experience that many are accustomed to. At the opposite end, AI driven research is looking at connecting the virtual world with the physical world we live in, seamlessly, breaking down the perception of privacy. This research does not attempt to define privacy-boundaries, which are left for behavioural research studies. Instead, this research presents underlying technologies, their implementation details, and their impact on user-privacy within a design science paradigm.

Alternative tracking techniques have been discussed widely in previous research studies in the past decade. Those tracking techniques have a wider relevance in Information Science and Business research disciplines where AM and other e-advertising and web traffic generation activities play a major role. Many research studies in those fields have been referring to indestructible tracking techniques from last decade. The first goal of this research was to evaluate if those techniques were still relevant and to update the current knowledge on alternative stateful tracking vectors. The efficacy of different tracking vectors was verified through experiments and HTML5 local storage and ETags were found to be still useful, while Flash cookies, Silverlight, the Super-cookie concept were already outdated.

The second goal was to use the alternative tracking vectors that are still relevant, to strengthen the HTTP cookie-based tracking technology. Experiments that used HTML5 Local storage and ETags as tracking vectors were designed to test nine predefined test scenarios and the results showed that both vectors outperformed the traditional HTTP cookie as a tracking vector. As ETags passed some tests that HTML5 Local storage failed and vice versa, an improved tracking capability was demonstrated by combining the three tracking vectors.

Though many previous research studies discuss the possibility of using alternative tracking vectors, implementation details of such techniques were not seen during the review of literature. Following a design science research paradigm, this research demonstrates how those tracking vectors can be used to strengthen underlying tracking technologies under a variety of e-commerce use cases.

This research has also demonstrated through experiments that the HTTP cookie-based tracking techniques are still the easiest to use, with least amount of extra code writing at software development stage and least resource intensive during program execution. As HTTP cookie is part of the HTTP protocol, it is well documented and will remain fit for the purpose for years to come. We have also demonstrated how HTML5 Local storage and ETag caching mechanism can be used to supplement the HTTP cookie-based tracking process, which can improve the reliability of the tracking capability, during some instances when HTTP cookies fail. We have also demonstrated that indestructible state of stateful tracking mechanisms, discussed in previous research literature, are not effective anymore. As these technologies were developed for purposes other than for user-tracking, their later updates and developments can render them useless for tracking purpose.

We have also demonstrated that many of these alternative tracking techniques, which did not adhere to privacy requirements, as mentioned in previous research literature, have over time, been implemented with the same restrictions by different browser manufacturers. For example, though many previous research studies have mentioned that HTML5 Local storage and ETags remain accessible even in incognito mode of the browser, we have found that browsers have by now already restricted access to both HTML5 Local storage and to ETags, during anonymous browsing mode.

Finally, the third goal was to improve the user-privacy during online tracking processes. The above experiments to improve the robustness of the tracking process were designed to use a non-PII based UID to represent users online. The experiments demonstrated how a user who is only known by a UID can be tracked across multiple domains, over time, fulfilling the tracking requirements, but still not

personally identifying the user. Using different web traffic generation models such as CPC, CPM and CPA, the web applications were able to successfully monitor user-activities such as banner advertisement clicks and purchase actions across multiple domains. They were able to attribute commission earnings to affiliates who generated that web traffic accurately, identifying the user only by the UID, without the need to know any personal information.

Despite demonstrating how the improved online tracking can be undertaken in a privacy-preserving manner, there are different service providers such as business analytics services and e-marketing services whose business models are based on acquiring large-scale personal and behavioural data of users, who have the choice to continue to gather data in a privacy-intrusive manner. Therefore, another set of experiments were designed to demonstrate how much personal information can be gathered based on the network reach associated with different tracking use-cases. A privacy model was developed based on the information seeking behaviour of the applications and resulting levels of privacy breaches. It presents a hierarchical view of privacy intruding use cases, which provides clarity to *web tracking*, a term that has over time become synonymous with stalking.

This knowledge will be useful for formulation and enacting of effective privacy legislations that protect privacy of individual users, but at the same time, does not curtail or hinder the development of web technologies.

As outputs of this Design Science research, implementation details for each alternative tracking vector are presented separately and in combination with other tracking vectors, which makes the process more robust. They are presented in chapter 4 *Artefact Description*, as sequence diagrams with additional process descriptions. The multi-domain test environment containing *Advertiser* domains, *Affiliate* domains and the *Tracking* domain has been instantiated as a functional prototype, which has been made publicly available for testing and demonstration.

7.1 Future direction

In this research we have investigated the use of different stateful tracking techniques. They provide high accuracy, as the identifier is stored within the client computer and accurate identification is as simple as reading the saved identifier. Such process has little computational processing overheads and is suitable for a production server that might handle large number of HTTP requests. Stateless tracking techniques have improved its accuracy of identification over time and might be a viable option to explore. It requires gathering of large number of features to create a unique footprint, which require more processing resources and may incur some latency. The viability of adding stateless tracking was not explored in this research due to the limitation of scope but would be considered in future research endeavours.

Federated Learning of Cohorts (FLoC) proposed by Google is a new way of targeting advertisements towards groups of people who share common interests without identifying individuals. It is an attempt to replace third-party cookies and identification of individuals online, while still being able to generate revenue through advertising (Chetna, 2021). It is still at experimental stage and has the potential to change direction of individual user tracking altogether. Nevertheless, though it would offer an alternative for digital marketing, the same technology might not replace the needs of other tracking purposes. Nevertheless, what impact FLoC has on the tracking requirements discussed in this research and how it might be incorporated will be an important consideration.

The *App Tracking Transparency* framework introduced by Apple Inc. since the release of iOS 14.5 in April 2021 restricts tracking by third-party phone Apps by default (Apple, 2021; Facebook, 2021a). While Facebook has been campaigning against Apple's new privacy feature, calling it a monopoly of advertising revenue, how far it restricts user identification during browsing for non-advertising purposes discussed in this research, is yet to be studied. The new feature blocks access to the system advertising identifier (IDFA) on the phone and other identifiers such as email address, but using other tracking vectors and stateless tracking techniques, need to be studied in future research.

Appendix



STATEMENT OF CONTRIBUTION DOCTORATE WITH PUBLICATIONS/MANUSCRIPTS

We, the candidate and the candidate's Primary Supervisor, certify that all co-authors have consented to their work being included in the thesis and they have accepted the candidate's contribution as indicated below in the *Statement of Originality*.

Name of candidate:	Bede Ravindra Amarasekara	
Name/title of Primary Supervisor:	Assoc. Prof. Anuradha Mathrani	
Name of Research Output and full reference:		
Crookies: Tampering with Cookies to Defraud E-Marketing		
In which Chapter is the Manuscript /Published work:	73	
Please indicate:		
<ul style="list-style-type: none"> The percentage of the manuscript/Published Work that was contributed by the candidate: 	80	
and		
<ul style="list-style-type: none"> Describe the contribution that the candidate has made to the Manuscript/Published Work: 	<p>The candidate is the lead author and drove this study. In this study, the candidate designed experiments for investigating different tracking tactics being used in e-marketing scenarios and which have led to identification of vulnerabilities existing in</p>	
For manuscripts intended for publication please indicate target journal:		
Published (http://doi:10.4018/978-1-5225-9715-5.ch073)		
Candidate's Signature:	Bede Ravindra Amarasekara	<small>Digitally signed by Bede Ravindra Amarasekara DN: cn=Bede Ravindra Amarasekara, o=NZ, email=bede@amarasekara.net Reason: I agree to the terms defined by the placement of my signature on this document Location: Auckland, New Zealand Date: 2021.05.29 13:45:16 +1200</small>
Date:	29/05/2021	
Primary Supervisor's Signature:	Anuradha Mathrani	<small>Digitally signed by Anuradha Mathrani DN: cn=Anuradha Mathrani, o=NZ, ou=Massey University, ou=School of Natural and Computational Sciences, email=A.S.Mathrani@massey.ac.nz Date: 2021.05.29 16:32:44 +1200</small>
Date:	29/05/2021	

(This form should appear at the end of each thesis chapter/section/appendix submitted as a manuscript/ publication or collected as an appendix at the end of the thesis)



STATEMENT OF CONTRIBUTION DOCTORATE WITH PUBLICATIONS/MANUSCRIPTS

We, the candidate and the candidate's Primary Supervisor, certify that all co-authors have consented to their work being included in the thesis and they have accepted the candidate's contribution as indicated below in the *Statement of Originality*.

Name of candidate:	Bede Ravindra Amarasekara	
Name/title of Primary Supervisor:	Assoc. Prof. Anuradha Mathrani	
Name of Research Output and full reference:		
Security and privacy management in cross-domain tracking systems within an e-marketing context		
In which Chapter is the Manuscript /Published work:		
Please indicate:		
<ul style="list-style-type: none"> The percentage of the manuscript/Published Work that was contributed by the candidate: 	85	
and		
<ul style="list-style-type: none"> Describe the contribution that the candidate has made to the Manuscript/Published Work: 		
The candidate is the lead author and drove this study. The candidate implemented a cross-domain tracking prototype for simulating different privacy and security breaches. This paper provides an in-depth description of the underlying		
For manuscripts intended for publication please indicate target journal:		
Published 9 https://doi.org/10.1109/CSDE48274.2019.9162393)		
Candidate's Signature:	Bede Ravindra Amarasekara	<small>Digitally signed by Bede Ravindra Amarasekara DN: cn=Bede Ravindra Amarasekara, o=NZ, email=bede@amarasekara.net Reason: I agree to the terms defined by the placement of my signature on this document Location: Auckland, New Zealand Date: 2021.05.29 13:58:01 +1200</small>
Date:	29/05/2021	
Primary Supervisor's Signature:	Anuradha Mathrani	<small>Digitally signed by Anuradha Mathrani DN: cn=Anuradha Mathrani, o=NZ, ou=Massey University, ou=School of Natural and Computational Sciences, email=A. S. Mathrani@massey.ac.nz Date: 2021.05.29 15:38:22 +1200</small>
Date:	29/05/2021	

(This form should appear at the end of each thesis chapter/section/appendix submitted as a manuscript/ publication or collected as an appendix at the end of the thesis)



STATEMENT OF CONTRIBUTION DOCTORATE WITH PUBLICATIONS/MANUSCRIPTS

We, the candidate and the candidate's Primary Supervisor, certify that all co-authors have consented to their work being included in the thesis and they have accepted the candidate's contribution as indicated below in the *Statement of Originality*.

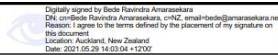
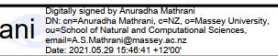
Name of candidate:	Bede Ravindra Amarasekara	
Name/title of Primary Supervisor:	Assoc. Prof. Anuradha Mathrani	
Name of Research Output and full reference:		
Improving the Robustness of the Cross-Domain Tracking Process		
In which Chapter is the Manuscript /Published work:		
Please indicate:		
<ul style="list-style-type: none"> The percentage of the manuscript/Published Work that was contributed by the candidate: 	90	
and		
<ul style="list-style-type: none"> Describe the contribution that the candidate has made to the Manuscript/Published Work: 	The candidate is the lead author and drove this study. The candidate designed many tracking techniques to improve the robustness of cookie-based tracking systems. This paper contributes to both theory and practice as it highlights how	
For manuscripts intended for publication please indicate target journal:		
Published (https://doi.org/10.1007/978-981-15-3380-8_23)		
Candidate's Signature:	Bede Ravindra Amarasekara	<small>Digitally signed by Bede Ravindra Amarasekara DN: cn=Bede Ravindra Amarasekara, o=NZ, email=bede@amarasekara.net Reason: I agree to the terms defined by the placement of my signature on this document Location: Auckland, New Zealand Date: 2021.05.29 13:57:38 +1200</small>
Date:	29/05/2021	
Primary Supervisor's Signature:	Anuradha Mathrani	<small>Digitally signed by Anuradha Mathrani DN: cn=Anuradha Mathrani, cn=NZ, o=Massey University, ou=School of Natural and Computational Sciences, email=A.S.Mathrani@massey.ac.nz Date: 2021.05.29 15:41:36 +1200</small>
Date:	29/05/2021	

(This form should appear at the end of each thesis chapter/section/appendix submitted as a manuscript/ publication or collected as an appendix at the end of the thesis)



STATEMENT OF CONTRIBUTION DOCTORATE WITH PUBLICATIONS/MANUSCRIPTS

We, the candidate and the candidate's Primary Supervisor, certify that all co-authors have consented to their work being included in the thesis and they have accepted the candidate's contribution as indicated below in the *Statement of Originality*.

Name of candidate:	Bede Ravindra Amarasekara	
Name/title of Primary Supervisor:	Assoc. Prof. Anuradha Mathrani	
Name of Research Output and full reference:		
Stuffing, Sniffing, Squatting, and Stalking: Sham Activities in Affiliate Marketing		
In which Chapter is the Manuscript /Published work:	6	
Please indicate:		
<ul style="list-style-type: none"> The percentage of the manuscript/Published Work that was contributed by the candidate: 	90	
and		
<ul style="list-style-type: none"> Describe the contribution that the candidate has made to the Manuscript/Published Work: 	<p>The candidate is the lead author and drove this study. The candidate has described managerial and technical interventions that can assist e-businesses in increasing their global visibility, while at the same time safeguarding their online trading</p>	
For manuscripts intended for publication please indicate target journal:		
Published (https://doi.org/10.1353/lib.2020.0016)		
Candidate's Signature:	Bede Ravindra Amarasekara 	<small>Digitally signed by Bede Ravindra Amarasekara DN: cn=Bede Ravindra Amarasekara, o=NC, email=bede@amarasekara.net Reason: I agree to the terms defined by the placement of my signature on this document Location: Auckland, New Zealand Date: 2021.05.29 14:03:04 +1200</small>
Date:	29/05/2021	
Primary Supervisor's Signature:	Anuradha Mathrani 	<small>Digitally signed by Anuradha Mathrani DN: cn=Anuradha Mathrani, o=NC, o=Massey University, ou=School of Natural and Computational Sciences, email=A.S.Mathrani@massey.ac.nz Date: 2021.05.29 15:46:41 +1200</small>
Date:	29/05/2021	

(This form should appear at the end of each thesis chapter/section/appendix submitted as a manuscript/ publication or collected as an appendix at the end of the thesis)



STATEMENT OF CONTRIBUTION DOCTORATE WITH PUBLICATIONS/MANUSCRIPTS

We, the candidate and the candidate's Primary Supervisor, certify that all co-authors have consented to their work being included in the thesis and they have accepted the candidate's contribution as indicated below in the *Statement of Originality*.

Name of candidate:	Bede Ravindra Amarasekara	
Name/title of Primary Supervisor:	Assoc. Prof. Anuradha Mathrani	
Name of Research Output and full reference:		
Online Tracking: When Does it Become Stalking?		
In which Chapter is the Manuscript /Published work:		
Please indicate:		
<ul style="list-style-type: none"> The percentage of the manuscript/Published Work that was contributed by the candidate: 	90	
and		
<ul style="list-style-type: none"> Describe the contribution that the candidate has made to the Manuscript/Published Work: 		
The candidate is the lead author and drove this study. This study provides rich descriptions of experimental findings from the candidate's study and examines how robust tracking mechanisms can be implemented without invading the privacy of		
For manuscripts intended for publication please indicate target journal:		
Published (https://doi.org/10.1142/S2196888821500226)		
Candidate's Signature:	Bede Ravindra Amarasekara	<small>Digitally signed by Bede Ravindra Amarasekara DN: cn=Bede Ravindra Amarasekara, o=NZ, email=bede@amarasekara.net Reason: I agree to the terms defined by the placement of my signature on this document Location: Auckland, New Zealand Date: 2021.05.29 14:08:57 +1200</small>
Date:	29/05/2021	
Primary Supervisor's Signature:	Anuradha Mathrani	<small>Digitally signed by Anuradha Mathrani DN: cn=Anuradha Mathrani, o=NZ, ou=Massey University, ou=School of Natural and Computational Sciences, email=A.S.Mathrani@massey.ac.nz Date: 2021.05.29 16:52:50 +1200</small>
Date:	29/05/2021	

(This form should appear at the end of each thesis chapter/section/appendix submitted as a manuscript/ publication or collected as an appendix at the end of the thesis)

References

- Adobe. (2015). *Adobe Flash Player - Local Settings Manager*.
https://help.adobe.com/archive/en_US/FlashPlayer/LSM/flp_local_settings_manager.pdf
- Adobe. (2020). *Adobe Flash Player EOL General Information Page*
<https://www.adobe.com/nz/products/flashplayer/end-of-life.html>
- Afriat, H., Dvir-Gvirsman, S., Tsuriel, K., & Ivan, L. (2021). "This is capitalism. It is not illegal": Users' attitudes toward institutional privacy following the Cambridge Analytica scandal. *Information Society*, 37(2), 115-127. <https://doi.org/10.1080/01972243.2020.1870596>
- Alabbas, A., & Bell, J. (2018). Indexed Database API 2.0. *W3C Recommendation*.
<https://www.w3.org/TR/IndexedDB-2/#privacy>
- Alabbas, A., & Bell, J. (2021). Indexed Database API 3.0. *W3C Public Working Draft*.
<https://www.w3.org/TR/IndexedDB/#user-tracking>
- Amarasekara, B. R., & Mathrani, A. (2015). *Exploring Risk and Fraud Scenarios in Affiliate Marketing Technologies from the Advertiser's perspective* Australasian Conference in Information Systems (ACIS2015), Adelaide. <https://arxiv.org/abs/1606.01428>
- Amarasekara, B. R., & Mathrani, A. (2016). Controlling Risks and Fraud in Affiliate Marketing: A Simulation and Testing Environment. PST2016 (Privacy, Security and Trust - IEEE 14th Annual Conference),
- Amarasekara, B. R., & Mathrani, A. (2017). Revenue fraud in e-commerce platforms: Challenges and solutions for affiliate marketing. In A. Colarik, J. Jang-Jaccard, & A. Mathrani (Eds.), *Cyber Security and Policy - A Substantive Dialogue* (pp. 67-86). Massey University Press.
<http://www.masseypress.ac.nz/books/all/all/cyber-security-and-policy>
- Amit, R., & Zott, C. (2001). Value creation in E-business. *Strategic Management Journal*, 22(6-7), 493-520.
- Andriamilanto, N., Allard, T., & Guelvouit, G. L. (2021). "Guess Who?" Large-Scale Data-Centric Study of the Adequacy of Browser Fingerprints for Web Authentication. In L. Barolli, A. Poniszewska-Maranda, & H. Park (Eds.), (Vol. 1195, pp. 161-172). Springer International Publishing.
- Apple. (2021). If an app asks to track your activity. Retrieved 19/05/2021, from
<https://support.apple.com/en-nz/HT212025>
- Asdemir, K., Kumar, N., & Jacob, V. S. (2012). Pricing models for online advertising: CPM vs. CPC. *Information Systems Research*, 23(3-part-1), 804-822.
- Ayenson, M. D., Wambach, D. J., Soltani, A., Good, N., & Hoofnagle, C. J. (2011). Flash Cookies And Privacy II: Now with HTML5 and ETag Respawning. <https://ssrn.com/abstract=1898390>
- Bakir, V. (2020). Psychological Operations in Digital Political Campaigns: Assessing Cambridge Analytica's Psychographic Profiling and Targeting. *Frontiers in Communication*, 5.
<https://doi.org/10.3389/fcomm.2020.00067>
- Banase, C., Herrmann, D., & Federrath, H. (2012). Tracking Users on the Internet with Behavioral Patterns: Evaluation of Its Practical Feasibility. In D. Gritzalis, S. Furnell, & M. Theoharidou, *Information Security and Privacy Research* Berlin, Heidelberg.
- Barth, A., & Berkeley, U. C. (2011). HTTP State Management Mechanism. *IETF Internet RFCs*, 6265.
<https://tools.ietf.org/html/rfc6265>
- Bath, A. (2011). The web origin concept. *IETF Internet RFCs*, 6454. <https://tools.ietf.org/html/rfc6454>
- Baumann, A., Haupt, J., Gebert, F., & Lessmann, S. (2019). The Price of Privacy: An Evaluation of the Economic Value of Collecting Clickstream Data. *Business & Information Systems Engineering*, 61(4), 413-431. <https://doi.org/10.1007/s12599-018-0528-2>

- Belloro, S., & Mylonas, A. (2018). I know what you did last summer: New persistent tracking mechanisms in the wild [Article]. *IEEE Access*, 6, 52779-52792. <https://doi.org/10.1109/ACCESS.2018.2869251>
- Benediktova, B., & Nevsad, L. (2008). *Affiliate Marketing - Perspective of content providers (Dissertation)* Department of Business Administration and Social Sciences, Lulea University of Technology]. <http://urn.kb.se/resolve?urn=urn:nbn:se:ltu:diva-58065>
- Benninger, C. (2006). AJAX Storage: A Look at Flash Cookies and Internet Explorer Persistence. *Foundstone Professional Services & Education, McAfee*. <https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.128.2523&rep=rep1&type=pdf>
- Berghel, H. (2018). Malice Domestic: The Cambridge Analytica Dystopia [Periodical]. *Computer*, 51(5), 84-89. <https://doi.org/10.1109/MC.2018.2381135>
- Bilge, L., Strufe, T., Balzarotti, D., & Kirda, E. (2009). *All your contacts are belong to us: automated identity theft attacks on social networks* Proceedings of the 18th international conference on World wide web, Madrid, Spain. <https://doi.org/10.1145/1526709.1526784>
- Boyd, D., & Hargittai, E. (2010). Facebook privacy settings: Who cares? *First Monday*, 15(8). <https://doi.org/10.5210/fm.v15i8.3086>
- Brear, D., & Barnes, S. J. (2008). Assessing the value of online affiliate marketing in the UK financial services industry. *International Journal of Electronic Finance*, 2(1), 1-17.
- Broekhuizen, T. L. J., Bakker, T., & Postma, T. J. B. M. (2018). Implementing new business models: What challenges lie ahead? *Business Horizons*, 61(4), 555-566.
- Brown, A. J. (2020). "Should I Stay or Should I Leave?": Exploring (Dis)continued Facebook Use After the Cambridge Analytica Scandal [Article]. *Social Media and Society*, 6(1). <https://doi.org/10.1177/2056305120913884>
- Buhov, D., Rauchberger, J., & Schrittwieser, S. (2018). FLASH: Is the 20th Century Hero Really Gone? Large-Scale Evaluation on Flash Usage & Its Security and Privacy Implications. *J. Wirel. Mob. Networks Ubiquitous Comput. Dependable Appl.*, 9(4), 26-40.
- Cahn, A., Alfeld, S., Barford, P., & Muthukrishnan, S. (2016). *An Empirical Study of Web Cookies* Proceedings of the 25th International Conference on World Wide Web, Montréal, Québec, Canada. <https://doi.org/10.1145/2872427.2882991>
- Canali, D., Bilge, L., & Balzarotti, D. (2014). *On the effectiveness of risk prediction based on users browsing behavior* Proceedings of the 9th ACM symposium on Information, computer and communications security, Kyoto, Japan. <https://doi.org.ezproxy.massey.ac.nz/10.1145/2590296.2590347>
- Castelluccia, C. (2012). Behavioural Tracking on the Internet: A Technical Perspective. In S. Gutwirth, R. Leenes, P. De Hert, & Y. Pouillet (Eds.), *European Data Protection: In Good Health?* (pp. 21-33). Springer Netherlands. https://doi.org/10.1007/978-94-007-2903-2_2
- Chachra, N., Savage, S., & Voelker, G. M. (2015). Affiliate Crookies: Characterizing Affiliate Marketing Abuse. IMC '15 Proceedings of the 2015 ACM Conference on Internet Measurement Conference, Tokyo, Japan.
- Chetna, B. (2021, 17/03/2021). Building a privacy-first future for web advertising. *Google Ads & Commerce Blog*. <https://blog.google/products/ads-commerce/2021-01-privacy-sandbox>
- Constantinides, P., Henfridsson, O., & Parker, G. G. (2018). Introduction—Platforms and infrastructures in the digital age. *Information Systems Research*, 29(2), 381-400.
- Dennis, L., & Duffy. (2005). Affiliate marketing and its impact on e-commerce. *Journal of Consumer Marketing*, 22(3), 161-163.
- Dwyer, C. (2009). *Behavioral Targeting: A Case Study of Consumer Tracking on Levis.com* AMCIS 2009 Proceedings, San Francisco, CA. <https://aisel.aisnet.org/amcis2009/460/>
- Eckersley, P. (2010). How unique is your web browser? In M. J. Atallah & N. J. Hopper (Eds.), *Privacy Enhancing Technologies* (Vol. 6205, pp. 1-18). Springer.
- Edelman, B. (2015). *Affiliate fraud litigation index*. [http:// www.benedelman.org/affiliate-litigation](http://www.benedelman.org/affiliate-litigation)

- Edelman, B., & Brandi, W. (2015). Risk, Information, and Incentives in Online Affiliate Marketing. *Journal of Marketing Research*, *LII*, 1-12.
- Englehardt, S., & Narayanan, A. (2016). Online tracking: A 1-million-site measurement and analysis. Proceedings of the ACM SIGSAC Conference on Computer and Communications Security, Vienna, Austria.
- Englehardt, S., Reisman, D., Eubank, C., Zimmerman, P., Mayer, J., Narayanan, A., & Felten, E. W. (2015). Cookies That Give You Away: The Surveillance Implications of Web Tracking. In *Proceedings of the 24th International Conference on World Wide Web* (pp. 289–299). International World Wide Web Conferences Steering Committee.
<https://doi.org/10.1145/2736277.2741679>
- Facebook. (2021a). *How does Facebook receive information from other businesses and organizations?* Retrieved 12/05/2021 from
https://www.facebook.com/help/fblite/2230503797265156?helpref=faq_content
- Facebook. (2021b). Off-Facebook Activity. Retrieved 13/05/2021, from
https://www.facebook.com/off_facebook_activity
- Faou, M., Lemay, A., Deary-Hetu, D., Calvet, J., Labreche, F., Jean, M., . . . Fernandez, J. M. (2016). Follow the traffic: Stopping click fraud by disrupting the value chain. In *14th Annual Conference on Privacy, Security and Trust (PST)* (pp. 464-474). IEEE.
- Feal, Á., Gamba, J., Vallina-Rodriguez, N., Wijesekera, P., Reardon, J., Egelman, S., & Tapiador, J. (2020). Don't accept candies from strangers: An analysis of third-party SDKs. In D. Hallinan, R. Leenes, & P. d. Hert (Eds.), *Data Protection and Privacy-Data Protection and Artificial Intelligence*. Bloomsbury.
- Fielding, R. (2014). Hypertext Transfer Protocol (HTTP/1.1): Semantics and Content. *IETF Internet RFCs*, 7231. <https://tools.ietf.org/html/rfc7231>
- Fielding, R., & Reschke, J. (2014). Hypertext Transfer Protocol (HTTP/1.1): Conditional Requests. *IETF Internet RFCs*, 7232. <https://tools.ietf.org/html/rfc7232>
- Gamba, J., Rashed, M., Razaghpanah, A., Tapiador, J., & Vallina-Rodriguez, N. (2020). An Analysis of Pre-installed Android Software. *2020 IEEE Symposium on Security and Privacy*, 1039-1055.
<https://doi.org/10.1109/SP40000.2020.00013>
- GDPR. (2016). *General Data Protection Regulation*. Retrieved from
<http://data.europa.eu/eli/reg/2016/679/oj>
- Goldkuhl, G. (2004). Design theories in information systems-a need for multi-grounding. *Journal of Information Technology Theory and Application (JITTA)*, *6*(2), 59-72.
- Google. (2021a). Manage your Location History. *Google Account Help*. Retrieved 7/05/2021, from
<https://support.google.com/accounts/answer/3118687?hl=en#>
- Google. (2021b). *Universal Analytics usage guidelines*. Google LLC. Retrieved 30/9/2019 from
<https://support.google.com/analytics/answer/2795983?hl=en>
- Gregor, S., & Hevner, A. R. (2013). Positioning and presenting design science research for maximum impact. *MIS Quarterly*, *37*(2), 337-355.
- Gregori, N., Daniele, R., & Altinay, L. (2013). Affiliate Marketing in Tourism: Determinants of Consumer Trust. *Journal of Travel Research*, *53*(2), 196-210.
- Hand, D. J. (2018). Aspects of data ethics in a changing world: Where are we now? *Big data*, *6*(3), 176-190.
- Harding, W. T., Reed, A. J., & Gray, R. L. (2001). Cookies and Web Bugs: What they are and how they work together. *Information Systems Management*, *18*(3), 17-24.
- Hevner, A. R., & Chatterjee, S. (2015). Design Science Research in Information Systems. In J. v. Brocke (Ed.), *Association for Information Systems Reference Syllabi*. Eduglopedia.org.
http://eduglopedia.org/reference-syllabus/AIS_Reference_Syllabus_Design_Science_Research_in_IS.pdf
- Hevner, A. R., March, S. T., Park, J., & Ram, S. (2004). Design Science In Information Systems Research. *MIS Quarterly*, *28*(1), 75-105.

- Hickson, I. (2010). Web SQL Database. W3C Working group. <https://www.w3.org/TR/webdatabase/>
- Hickson, I. (2021). W3C Recommendation - Web Storage. W3C working Group. <https://www.w3.org/TR/webstorage/#privacy>
- Hinds, J., Williams, E. J., & Joinson, A. N. (2020). "It wouldn't happen to me": Privacy concerns and perspectives following the Cambridge Analytica scandal. *International Journal of Human - Computer Studies*, 143. <https://doi.org/10.1016/j.ijhcs.2020.102498>
- Holmström, J., Ketokivi, M., & Hameri, A.-P. (2009). Bridging Practice and Theory: A Design Science Approach [Article]. *Decision Sciences*, 40(1), 65-87. <https://doi.org/10.1111/j.1540-5915.2008.00221.x>
- Hoofnagle, C. J., Urban, J., & Li, S. (2012). Privacy and Modern Advertising: Most US Internet Users Want 'Do Not Track' to Stop Collection of Data about their Online Activities. Amsterdam Privacy Conference, Amsterdam, Netherlands.
- Hu, J., Zeng, H.-J., Li, H., Niu, C., & Chen, Z. (2007). *Demographic prediction based on user's browsing behavior* Proceedings of the 16th international conference on World Wide Web, Banff, Alberta, Canada. <https://doi-org.ezproxy.massey.ac.nz/10.1145/1242572.1242594>
- Hu, Y., Shin, J., & Tang, Z. (2013). Performance-based Pricing Models in Online Advertising: Cost per Click versus Cost per Action. <http://spinup-000d1a-wp-offload-media.s3.amazonaws.com/faculty/wp-content/uploads/sites/32/2019/06/Onlineadvertising2013.pdf>
- Ismagilova, E., Slade, E. L., Rana, N. P., & Dwivedi, Y. K. (2020). The effect of electronic word of mouth communications on intention to buy: a meta-analysis. *Information Systems Frontiers*, 22(5), 1203-1226. <https://doi.org/10.1007/s10796-019-09924-y>
- Kayalvizhi, R., Khattar, K., & Mishra, P. (2018). A Survey on Online Click Fraud Execution and Analysis. *International Journal of Applied Engineering Research*, 13(18), 13812-13816.
- Kilubi, I. (2015). Strategic technology partnering: A framework extension. *The Journal of High Technology Management Research*, 26(1), 27-37.
- Krishnamurthy, B., & Wills, C. E. (2009). Privacy diffusion on the web: A longitudinal perspective. *Proceedings of the 18th International Conference on World Wide Web*, 541-550. <https://doi.org/10.1145/1526709.1526782>
- Kristol, D. M. (2001). HTTP Cookies: Standards, Privacy, and Politics. *ACM Transactions on Internet Technology*, 1(2), 151-198.
- Kristol, D. M., & Montulli, L. (1997). HTTP State Management Mechanism. *IETF Internet RFCs*, 2109. <https://tools.ietf.org/html/rfc2109>
- Kuechler, B., & Vaishnavi, V. (2008). On theory development in design science research: anatomy of a research project. *European Journal of Information Systems*, 17(5), 489-504. <https://doi.org/10.1057/ejis.2008.40>
- Laperdrix, P., Rudametkin, W., & Baudry, B. (2016). Beauty and the Beast: Diverting modern web browsers to build unique browser fingerprints. *37th IEEE Symposium on Security and Privacy*, 878-894. <https://doi.org/doi:10.1109/SP.2016.57>
- Laterza, V. (2018). Cambridge Analytica, independent research and the national interest [Editorial]. *Anthropology Today*, 34(3), 1-2. <https://doi.org/10.1111/1467-8322.12430>
- Libert, T. (2015). Exposing the Invisible Web: An Analysis of Third-Party HTTP Requests on 1 Million Websites. *International Journal Of Communication*. <https://ssrn.com/abstract=2685330>
- Logan, B. M., & Mossing, T. C. (2007). *Method, apparatus and computer program product for automatic cookie synchronization between distinct web browsers* (US Patent No. US20070157304A1).
- Manokha, I. (2018). Surveillance: The DNA of Platform Capital—The Case of Cambridge Analytica Put into Perspective [Article]. *Theory & Event*, 21(4), 891-913.
- March, S. T., & Smith, G. F. (1995). Design and natural science research on information technology. *Decision Support Systems*, 15(4), 251-266. [https://doi.org/10.1016/0167-9236\(94\)00041-2](https://doi.org/10.1016/0167-9236(94)00041-2)

- Margaret, H. (2020). Cambridge Analytica's black box. *Big Data & Society*, 7(2).
<https://doi.org/10.1177/2053951720938091>
- 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.
- Martin, D., Wu, H., & Alsaïd, A. (2003). Hidden surveillance by Web sites: Web bugs in contemporary use. *Commun. ACM*, 46(12), 258–264. <https://doi.org/10.1145/953460.953509>
- Matte, C., Bielova, N., & Santos, C. (2020, 18-21 May 2020). Do Cookie Banners Respect my Choice? : Measuring Legal Compliance of Banners from IAB Europe's Transparency and Consent Framework. 2020 IEEE Symposium on Security and Privacy (SP),
- Mayer, J. R., & Mitchell, J. C. (2012). Third-party web tracking: Policy and technology [Conference Paper]. *IEEE Symposium on Security and Privacy*, 413-427.
<https://doi.org/10.1109/SP.2012.47>
- McKnight, H. D., Choudhury, V., & Kacmar, C. (2002). Developing and Validating Trust Measures for e-Commerce: An Integrative Typology. *Information Systems Research*, 13(3), 334-359.
- Microsoft. (2020). *Silverlight End of Support* <https://support.microsoft.com/en-us/windows/silverlight-end-of-support-0a3be3c7-bead-e203-2dfd-74f0a64f1788>
- Mittal, S. (2010). *User Privacy and the Evolution of Third-party Tracking Mechanisms on the World Wide Web*. Stanford University.
https://stacks.stanford.edu/file/druid:hw648fn9717/SonalMittal_Thesis.pdf
- Mor, N., Riva, O., Nath, S., & Kubiatoiwicz, J. (2015). Bloom Cookies: Web Search Personalization without User Tracking. *22nd Annual Network and Distributed System Security Symposium*.
<https://doi.org/10.14722/ndss.2015.23108>
- Narayanan, A., & Reisman, D. (2017). The Princeton web transparency and accountability project. In Cerquitelli T., Quercia D., & P. F. (Eds.), *Transparent Data Mining for Big and Small Data* (Vol. 32, pp. 45-67). Springer. https://doi.org/https://doi.org/10.1007/978-3-319-54024-5_3
- Norouzi, A. (2017). An Integrated survey in Affiliate Marketing Network [Paper presentation]. *2nd World Conference on Technology, Innovation and Entrepreneurship*, 42, 299-309.
<https://doi.org/10.17261/Pressacademia.2017.604>
- Nunamaker, J. F., Chen, M., & Pruding, T. D. M. (1991). Systems Development in Information Systems Research. *Journal of Management Information Systems*, 7(3), 89-106.
<https://doi.org/10.1080/07421222.1990.11517898>
- O'Brien, P., Young, S. W. H., Arlitsch, K., & Benedict, K. (2018). Protecting privacy on the web : A study of HTTPS and Google Analytics implementation in academic library websites. *Online Information Review*, 42(6), 734-751. <https://doi.org/10.1108/OIR-02-2018-0056>
- Olbrich, R., Bormann, P. M., & Hundt, M. (2019). Analyzing the Click Path Of Affiliate-Marketing Campaigns: Interacting Effects of Affiliates' Design Parameters With Merchants' Search-Engine Advertising [Article]. *Journal of Advertising Research*, 59(3), 342-356.
<https://doi.org/10.2501/JAR-2018-043>
- Olejnik, L. (2019). W3C TAG Observations on Private Browsing Modes. *W3C Tag Finding*.
<https://www.w3.org/2001/tag/doc/private-browsing-modes/>
- Papadogiannakis, E., Papadopoulou, P., Kourtellis, N., & Markatos, E. P. (2021). User Tracking in the Post-cookie Era: How Websites Bypass GDPR Consent to Track Users [Paper presentation]. *WWW '21: Proceedings of the Web Conference 2021*, 2130-2141.
<https://doi.org/10.1145/3442381.3450056>
- Pawan, K., & Gursimranjit, S. (2020). Using Social Media and Digital Marketing Tools and Techniques for Developing Brand Equity With Connected Consumers. In D. Sumesh Singh (Ed.), *Handbook of Research on Innovations in Technology and Marketing for the Connected Consumer* (pp. 336-355). IGI Global. <https://doi.org/10.4018/978-1-7998-0131-3.ch016>
- Peffer, K., Tuunanen, T., Rothenberger Marcus, A., & Chatterjee, S. (2007). A Design Science Research Methodology for Information Systems Research. *Journal of Management Information Systems*, 24(3), 45-77. <https://doi.org/10.2753/MIS0742-1222240302>

- Persily, N. (2017). Can Democracy survive the internet? *Journal of Democracy*, 28(2), 63-76.
- Ransbotham, S., Fichman, R. G., Gopal, R., & Gupta, A. (2016). Special section introduction— ubiquitous IT and digital vulnerabilities. *Information Systems Research*, 27(4), 834-847.
- Richterich, A. (2018). How Data-driven research fuelled the Cambridge Analytica controversy. *The Open Journal of Sociopolitical Studies*, 11(2), 528-543.
- Roesner, F., Kohno, T., & Wetherall, D. (2012). *Detecting and defending against third-party tracking on the web* [Paper presentation]. 9th {USENIX} Symposium on Networked Systems Design and Implementation ({NSDI} 12), San Jose, CA.
<https://www.usenix.org/conference/nsdi12/technical-sessions/presentation/roesner>
- Roosendaal, A. (2012). We Are All Connected to Facebook... by Facebook! In S. Gutwirth, R. Leenes, P. De Hert, & Y. Poullet (Eds.), *European Data Protection: In Good Health?* (pp. 3-19). Springer. https://doi.org/10.1007/978-94-007-2903-2_1
- Rutz, O. J., & Bucklin, R. E. (2007). A Model of Individual Keyword Performance in Paid Search Advertising. <https://doi.org/10.2139/ssrn.1024765>
- Sanchez-Rola, I., Dell'Amico, M., Balzarotti, D., Vervier, P.-A., Bilge, L., Hassan, W. U., . . . Han, Y. (2021). Journey to the Center of the Cookie Ecosystem: Unraveling Actors' Roles and Relationships. *Proceedings of the IEEE Symposium on Security and Privacy (S&P)*,
- Sanchez-Rola, I., Dell'Amico, M., Kotzias, P., Balzarotti, D., Bilge, L., Vervier, P.-A., & Santos, I. (2019). *Can I Opt Out Yet? GDPR and the Global Illusion of Cookie Control* *Proceedings of the 2019 ACM Asia Conference on Computer and Communications Security*, Auckland, New Zealand.
<https://doi-org.ezproxy.massey.ac.nz/10.1145/3321705.3329806>
- Sanchez-Rola, I., & Santos, I. (2018). Knockin' on trackers' door: Large-scale automatic analysis of web tracking. In c. Giuffrida, S. Bardin, & G. Blanc (Eds.), *International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment* (Vol. 10885, pp. 281-302). Springer, Cham.
- Schelter, S., & Kunegis, J. (2016). *Tracking the trackers: a large-scale analysis of embedded web trackers* [Paper presentation]. AAAI International Conference on Weblogs and Social Media, Cologne, Germany. <https://ojs.aaai.org/index.php/ICWSM/article/view/14769>
- Sivakorn, S., Polakis, I., & Keromytis, A. D. (2016). The cracked cookie jar: HTTP cookie hijacking and the exposure of private information. *2016 IEEE Symposium on Security and Privacy (SP)*,
- Snyder, P., & Kanich, C. (2015). *No Please, After You: Detecting Fraud in Affiliate Marketing Networks* [Paper presentation]. Workshop on the Economics of Information Security (WEIS),
https://www.cs.uic.edu/pub/Bits/PeterSnyder/Snyder_Kanich_-_Cookie_Stuffing.pdf
- Snyder, P., & Kanich, C. (2016). Characterizing fraud and its ramifications in affiliate marketing networks. *Journal of Cybersecurity*, 2(1), 71-81.
- Soltani, A., Canty, S., Mayo, Q., Thomas, L., & Hoofnagle, C. J. (2010). *Flash Cookies and Privacy* [Paper presentation]. AAAI Spring Symposium: Intelligent Information Privacy Management, Palo Alto, California.
<https://www.aaai.org/ocs/index.php/SSS/SSS10/paper/viewPaper/1070>
- Starov, O., & Nikiforakis, N. (2018). Privacymeter: Designing and developing a privacy-preserving browser extension. In M. Payer, A. Rashid, & J. Such (Eds.), *International Symposium on Engineering Secure Software and Systems* (pp. 77-95). Springer, Cham.
https://doi.org/10.1007/978-3-319-94496-8_6
- Suryanarayana, S. A., Sarne, D., & Kraus, S. (2019). Information disclosure and partner management in affiliate marketing. In *Proceedings of the First International Conference on Distributed Artificial Intelligence* (pp. 1-8). ACM. <https://doi.org/10.1145/3356464.3357703>
- ur Rehman, I. (2019). Facebook-Cambridge Analytica data harvesting: What you need to know. *Library Philosophy and Practice*, 2019, 1-11.
<https://digitalcommons.unl.edu/libphilprac/2497>

- Utz, C., Degeling, M., Fahl, S., Schaub, F., & Holz, T. (2019). (Un)informed Consent: Studying GDPR Consent Notices in the Field. ACM SIGSAC Conference on Computer and Communications Security (CCS'19), London, United Kingdom.
- Wachter, S., & Mittelstadt, B. (2019). A Right to Reasonable Inferences: Re-Thinking Data Protection Law in the Age of Big Data and AI. *Colombia Business Law Review*, 2019(2), 130.
https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3248829#
- Wang, Y. (2018). Research on the Targeted Advertising of Agricultural Products Based on Flash Cookie Technology. *DEStech Transactions on Engineering and Technology Research International Conference on Mechanical, Electronic and Information Technology (ICMEIT 2018)*, Shanghai, China.
- Wilson, T. D. (1999). Models in information behaviour research. *Journal of documentation*, 55(3), 249-270.
- Xie, Y., Yu, F., Achan, K., Gillum, E., Goldszmidt, M., & Wobber, T. (2007). How dynamic are IP addresses? In *Proceedings of the 2007 conference on Applications, technologies, architectures, and protocols for computer communications* (pp. 301-312). ACM.
<https://doi.org/10.1145/1282380.1282415>
- Yang, Z., & Yue, C. (2020). A Comparative Measurement Study of Web Tracking on Mobile and Desktop Environments. *Proceedings on Privacy Enhancing Technologies, 2020*, 24-44.
<https://doi.org/10.2478/popets-2020-0016>