INSTITUTO SUPERIOR DE ENGENHARIA DO PORTO
MESTRADO EM ENGENHARIA INFORMÁTICA

Who is Tracking Me on the Web?

CARLOS MIGUEL GONÇALVES DA ROCHA
Outubro de 2016

POLITÉCNICO
DO PORTO

Who is Tracking Me on the Web?

Carlos Miguel Gonçalves Rocha

Dissertation to obtain the Master's Degree in Computer Engineering, Specializing in Computational Systems

Advisors:

Professor Nuno Pereira

Professor António Costa

Porto, October 2016

Dedication

I dedicate my dissertation to my family and many friends. A special feeling of gratitude to my loving parents, Graça and Carlos whose words of encouragement and push for tenacity ring in my ears to this day. My brother, Pedro, who always helped me with finding the correct word and sentence.

I also dedicate this dissertation to my friends who encouraged me not to give up and pushing forward, further helped me proof reading and provided me moments of fun when needed.

Finally, work colleagues who supported me throughout this long year by helping me with the work load and schedule.

Abstract

Lately, online privacy is an issue that has been gaining the focus of the media because more and more companies have adopted business models which offer its customers free services and increasing its value by manipulating and possibly selling the information acquired by gathering the data from its users on the web. These methods are diverse and have benefited from the technology advancements in the internet browsers and the evolution of technologies such as HTLM5 and JavaScript.

This dissertation aims to gather and analyze the information regarding the tracking that takes place in the Portuguese online market, this will be supported by a detailed state of the art of the technologies and methods used in the gathering of user information.

To identify the major players in web tracking the study will analyze the websites which Portuguese users frequently visit and evaluate their prevalence and the methods used to gather the information. With the goal of identifying those entities.

With this research, the author pretends to inform the public of who are the main entities involve in the process of collecting personal information, its methods used and how to prevent it.

A Paper based on a subset of 300 websites using the same framework and analysis tool of this research was published for the symposium INForum 2016, which took place in September of 2016.

Key-words: Privacy; Methods of gathering information; Entities;

Resumo

A privacidade na internet têm sido um tópico que nos últimos anos tem vindo a ganhar atenção dos média. Cada vez mais empresas adotam modelos de negócios que oferecem serviços gratuitos aos utilizadores, gerando valor por tratar e possivelmente vender informação adquirida pela recolha de dados da utilização dos *websites*. Os métodos de recolha são variados e tiram proveito dos avanços tecnológicos dos *browsers* e das tecnologias à sua volta, por exemplo HTLM5 e JavaScript.

Esta dissertação pretende reunir e analisar a informação relativa as tecnologias usadas para a recolha informação no mercado *online* português, vai ser suportado por um detalhado estado da arte das tecnologias e métodos usados na recolha de informação dos utilizadores.

Para identificar as principais entidades envolvidas na perda de privacidade dos utilizadores, este estudo vai analisar os *websites* que os portugueses mais frequentam e avaliar a prevalência dessas entidades e os métodos por esses usados.

Com esta investigação pretende-se informar as pessoas de quem são as principais entidades envolvidas nos processos de recolha de informação pessoal, os métodos usados e como o prevenir.

Um artigo baseado num subconjunto de 300 *websites* utilizando a mesma *framework* e mesma ferramenta de analise foi publicado para o simpósio INForum 2016 que se realizou em setembro de 2016.

Palavras-chaves: Privacidade; Metodologias de recolha de informação; Entidades;

Acknowledgements

I wish to start to thank my advisers, Professor Nuno Pereira and Professor António Costa, for their help throughout the initial phase, the writing of the paper published as part of my thesis work, and the final dissertation manuscript. Additionally, both allowed this paper to be my own work, but steered me in the right direction whenever they thought I needed it.

I would also like to thank the experts who were involved in the development of the framework used for this research: Steven Englehardt and Arvind Narayanan from Princeton University. Also, a word of appreciation to all the other users who helped to maintain and improve the framework throughout the years.

Table of Contents

List of Figures

List of Tables

Acronyms

Acronyms List

ABP	Adblock Plus
AJAX	Asynchronous JavaScript and XML
EFF	Electronic Frontier Foundation
HTTP	Hypertext Transfer Protocol
JDBC	Java Database Connectivity
LSO	Local Storage Objects
OS	Operating System
SQL	Structured Query Language
ToS	Terms of Service
URL	Uniform Resource Locator
WPM	Web Privacy Measurement

1 Introduction

Personal data is fueling a fast emerging industry which transforms the data into added value (Castelluccia *et al.* 2013), this gathering of personal data has become one of the pillars of the online economy and has been increasing as more people use the Internet.

The growth of online data collection has become a serious privacy concern because companies insert tracking code into a large number of web pages and use said data to create global view of user behavior (Datta *et al.* 2015).

Harvesting data is mostly done by systems, such as search engines, social networks, storage clouds, among others, where people provide their personal data in exchange for a service described as free.

In this chapter, it will be presented an overview of the problem that this thesis will try to provide an answer to, the motivation for this research and at the end it will be presented the structure of the document.

1.1 Problem

Web Tracking is a phenomenon which is predominant in today's web, as more advanced methods to uniquely identify users become available and associating these methods with the proliferation of cookies allows third-party services to accurately pin point what the user browses cross website and even cross-session.

The Portuguese market is rather small but very adapt to the international trends, but due to its small number of potential affected users there aren't any studies regarding the methods Portuguese websites use to track their users.

When talking about privacy is normal to mention “first-parties” which are the web-sites where the user is actively interacting with and it’s aware of its existence. Besides those, there is, most of the times, another party involved which are responsible for the hidden trackers embedded in most web pages from the ad-networks which are normally labeled “third-parties”.

The user has no control over these “third-parties” which leaves several open questions:

Who are they? Where they are? How can it be avoided?

Focusing on the Portuguese market, the goal is to gather a significant sample of websites and identify what information is being sent and where to, with this, it will be possible show a clear picture of the companies which gather user information.

Identifying the websites which contain this third-parties is an important step in raising awareness on the issue of online privacy, only by seeing the most of the user’s favorite websites contain mechanisms to track them, will the users understand how one company can create a complete profile of their online habits.

Stopping web tracking is not an easy task because tracking mechanism are associated with the core functionalities of the browsers, as such, stopping it will require either a change in third-parties, by respecting users wish to not be tracked or the user will have to sacrifice some browser functionalities in order to minimize tracking.

1.2 Context

At present time, it’s available several methods and techniques to track users online, and this presents serious challenges to an important security requirement: privacy, the possibility users have to hide their identity or information about their identity.

In the recent past, web tracking studies have presented evidence of sophisticated tracking methods in the wild, which exploit particular browser features to evade the tracking preferences of users (Acar *et al.* 2014). Another study has shown how web tracking can be leveraged by an adversary to perform mass surveillance, and identified frequent leakage of information that allowed identifying logged-in users (Englehardt *et al.* 2014). Companies which perform this type of tracking are commonplace nowadays and are part of what is called “third-parties.”

To gather the information, the most common and simple method is cookies but there are much more advance techniques such as web fingerprinting. This approach has different degrees of efficiency but the most sophisticated can uniquely identify a device by its browser, settings, operation system and even hardware.

Recognizing the widespread and abusive use of tracking mechanisms has already lead regulations of the European Commissions’ regarding privacy (e.g. (Bernal 2013)) and data

protection requiring user consent about the usage of tracking mechanisms (European Parliament 2012), and these have recently received increased attention due to the clarification of the regulations in light of attempts to circumvent user consent rules (Anon 2014).

1.3 Motivation, Methods and Objectives

The internet is a network of content available to anyone who has internet access. The content available takes many forms: online shopping, constant news updates and email providers, just to name a few examples. This content is provided to the users, most of the times, without charging money directly, this is possible due to the way the online economy works.

Most websites which users visits daily are embedded with third-party tracking mechanisms that are hidden from them. The third-parties present in the websites are capable of gathering an enormous amount of the user private online data, for example sites which he repeatedly visits and his online shopping habits. The entities behind the third-parties use this information to generate money without the user consent.

One of the main goals of this research is to bring to attention of the online users which are the most common third-parties and which tools they use to gather information. Similar studies have already been made but focused on bigger markets, however, this study will try to view the 'Portuguese internet' as a whole and identify which are the most common third-parties the users crosses in their daily online life.

To gather that information, it will be used the framework OpenWPM which was the base for several completed studies on the subject of web privacy. The tool was designed by researchers in Stanford University to provide reproducibility which gives the possibility to compare results and eventually standardize the web privacy measurements. This standardization is one of the main goals of this open-source framework.

The objectives of this research are:

- Which are the most predominant third-parties?
- What do the third-parties do with the information gathered?
- In which websites are third-parties present on?
- How does the cookie tracking work?
- How can third-parties be avoided?

Providing a detailed answer with evidence to each of this questions is the drive of this dissertation.

1.4 Results Achieved

From the list of the 700 most visited websites in Portugal, which contains 196 websites with the Portuguese top-level domain, each was visited and the cookies created upon their visit were analyzed without any privacy setting enable. This process was repeated two more times one with the extension Adblock Plus enabled and the other with extension Ghostery enabled.

The numbers of cookies created were reduced by 59,35% with ADP and by 84,02% with Ghostery. The number of third-party cookies also differed a lot when comparing the three executions, Adblock Plus reduced it by 75,14% and Ghostery reduced the number to 92,87%.

Finally, regarding tracking, cookies which share the same Host, Key and Value cross websites, the crawl with the extension ADP reduced the number of those cookies created in 72,24% while Ghostery managed to reduce it by 94,75%.

Should also be mentioned, the difference between ADP and Ghostery is the first is an advertisement blocker while the latter is a tracking blocker. This justifies why Ghostery has, overall, better results than ADP.

1.4.1 Paper

The 8th edition of INForum took place in Lisbon between 8 and 9 of September of 2016 and was organized by *Instituto Superior Técnico* (IST) and by INESC ID *Lisboa*.

Based on the data provided by this research a paper entitled “Web Tracking and Third-parties of Top Visited Domains in Portugal” was submitted to the approving committee by 20 of June with the help of both Professor Nuno Pereira and Professor António Costa.

The paper was based on a sample of the 300 most visited domains in Portugal, aimed at uncovering the techniques used for web tracking and identifying trackers which the Portuguese users are exposed to.

The paper was presented in Lisbon in September 2016.

1.5 Structure

The structure of this document is divided in six chapters: Introduction, State of the Art, Solution, Method, Results followed by a conclusion.

Chapter 1 (**Introduction)** focuses on the problem interpretation with an analysis of the motivation and the goals behind this research, a small overview of the methods used, the results obtained and a brief note regarding the paper published based on the work done in this study

In Chapter 2 (**State of the Art**), there is a detailed analysis of some methods used in web tracking providing context to the following chapters. It will also contains information on existing tools developed to minimize the effects of tracking. Additionally, it will be detailed the tools available to measure privacy.

The **Solution**, Chapter 3, is an in-depth analysis of the framework used and the tool developed to analyze and organize the raw data provided by the framework.

In the Chapter 4 (**Method**) of this document, there is a detailed overview of the process necessary to start the research, as well as, an analysis of the reliability of the framework.

On Chapter 5 (**Results**), it is illustrated with charts and tables the result of this research on the topics of cookies distribution, the identification of third-party, and the websites they appear on and how tracking is done.

Lastly, in the **Conclusion** it is detailed the results obtained, a list of recommendations based on the results, the answer to the questions behind this research and what can be done to further analyze this subject.

2 State of the Art

This chapter describes the existing tools and frameworks which are capable of tracking users and the tools suited to prevent online tracking. Additionally, it will also be detailed the tools and possible ways to measure online tracking.

In section 2.1 it will provide a sample of tracking methods which could be used by third-parties and details how they operate, 2.2 will provides a list of tools available to users which are capable of preventing some sorts of tracking, lastly, 2.3 are the methods developed to measure the tracking done by third-parties.

2.1 Tracking Mechanisms

The topic details some tools available to third-parties to track users on the web. From the list, some were already used, such as Evercookies and Canvas Fingerprinting, others are more theoretical which means they could be used but were identified in researches, for example, JavaScript Performance and No Script Whitelist.

2.1.1 Evercookies

Traditional browser cookies are the base of web tracking by its ability of storing state of the user on the local machine. A cookie is a triple (domain, key, value) stored in the browser across page visits, where domain is a web site, and key and value are opaque identifiers (Roesner *et al.* 2012).

Cookies are pulled either by JavaScript running in the page using an API call, or alternatively by HTTP responses which includes a Set-Cookie header. The browser attaches the cookies to a domain from an outgoing HTTP requests to the domain, using Cookie headers.

However, users began to be more aware of the presence of this files in their systems and understanding what they could be used for with the help of press. Being informed users, they started to delete the cookies from their computers due to the privacy issues associated with them.

Evercookies are designed to overcome the "shortcomings" of the traditional tracking mechanisms. By utilizing multiple storage vectors which are less transparent to users and may be more difficult to clear (Acar *et al.* 2014).

Plugins like Flash are a commonly used mechanism to allow websites to store data on the user's machine. In the case of Flash, websites can set Local Storage Objects (also referred to as "Flash cookies") on the user's file system (Roesner *et al.* 2012). Additionally, one of advances of HTML5 over its precedent was a new client-side storage mechanism for the web pages. Notably, the Local Storage which provides an API accessible storage area in which sites can use to hold information.

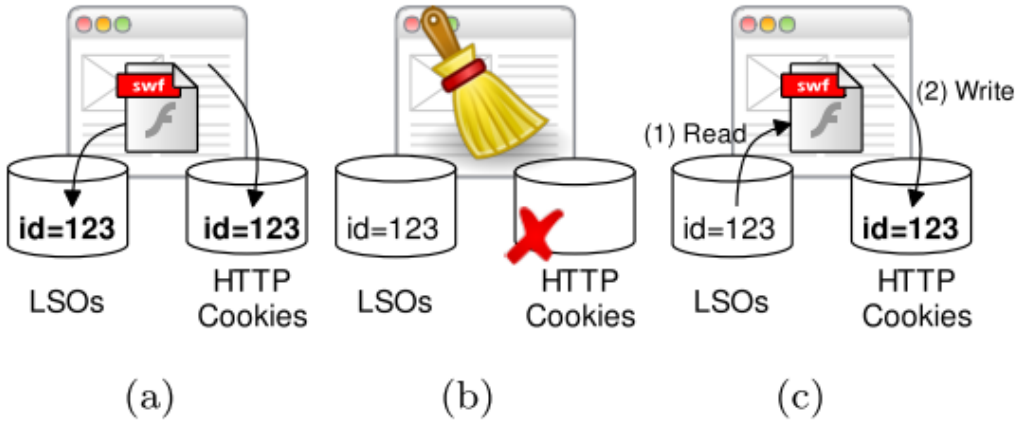

Figure 1 – Evercookies behavior (Acar *et al.* 2014)

Evercookies start like normal cookies, when a site is visited it creates a cookie in the target system but instead of just creating one, it uses all methods available like the HTML5 local storage and or the flash LSO, Local Storage Objects (a). The user is unaware of the other cookies created and at the end of its browsing session deletes the cookies (b). When the user revisits the site, instead of creating a new cookie it checks through HTML5 API's or Flash methods for the previous cookie, and recreates it (c), meaning the information stored in the cookies is never deleted even when the user actively tried to remove it.

2.1.2 Cookie Syncing

Cookie syncing allows two different third-parties to link their pseudonymous cookie IDs of the same user by including the cookie ID as a parameter of an URL, which is a redirection to the other third-party (Englehardt *et al.* 2014).

This process is part of Real-Time Bidding in which a third-party sells their cookie ID to a second third-party, this allows the buyer to have access to the info gathered by cookie of the seller.

Ad Exchange typically sends a script or a redirect instruction to instruct the user's browser to load a URL provided by the buyer with the Ad Exchange's user's cookie/id in the parameter. The buyer obtains the Ad Exchange's cookie/id upon receiving this request and matches this cookie/id with its own cookie (Ghosh & Roth 2013).

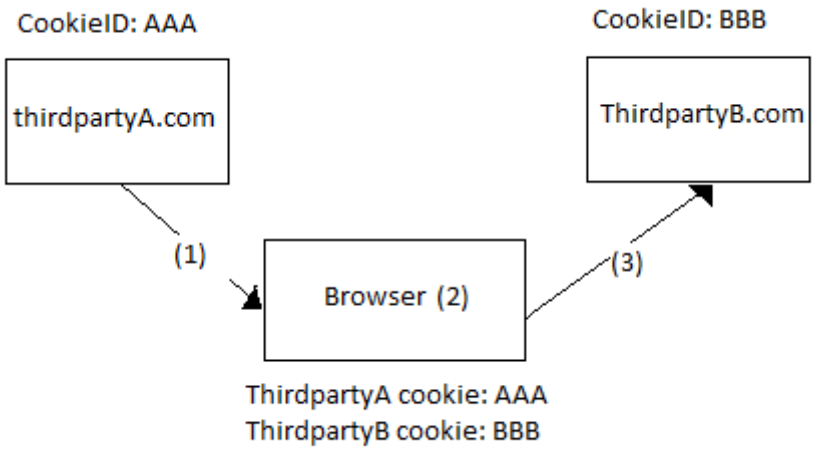

Figure 2 – Cookie Syncing process

In Cookie Syncing, one domain synchronizes their cookie with another domain by including it in the request sent to the latter. For example, a cookie created by the third-party A will send a request to third-party B with the cookie ID of third-party A (1) to the user browser, which will execute the server order (2), then third-party B will link both cookies (3). This happens when the user is vising a first party website.

Thus, to detect this it's needed to control the connections made from the third-parties embedded in the website to other third-parties. However, this approach doesn't work for every third-party, for example DoubleClick.

2.1.3 Web Fingerprinting

Unlike the common tracking methods such as unique cookie ID's and IP addresses, the Web or Browser Fingerprinting is a method or a collection of methods that uses the browser, operating system and hardware to uniquely identify users who visit the websites.

Web Fingerprinting has been the target of several researches and proposed implementations, some not very effective and others effective in uniquely identifying a user or his device. This kind of fingerprinting is both challenging to detect and difficult to prevent.

Upon visiting a web page, the user is at risk of being fingerprinted, by the information it sends to the server and the JavaScript being executed on the page. This JavaScript can look to the list of plug-ins installed, if Flash or Java is installed and/or enabled.

There are ways in which fingerprinting a user device may have positive effect, for example online banking, where this could be an additional layer of authentication, without the inconvenience of Security Tokens, with the assumption the user would login from the same device sites could use JavaScript fingerprint to prevent unauthorized logins with reduced user inconvenience (Mowery *et al.* 2011).

Unfortunately, the same method could be used to track users cross domains and be used to delay the detection of certain vulnerabilities by targeting only vulnerable system configurations (Mowery & Shacham 2012).

Bellow there are four implementations of finger printing processes.

HTTP Headers

Peter Eckersley from Electronic Frontier Foundation presented a simple method of fingerprinting a browser by checking the HTTP Headers of an HTTP request, more specifically, the User-Agent Field in the header. This field contains information regarding which browser is being used, its version number and details about the system, such as operating system and its version.

This method was tested in EEF domain, which means the data sample was bias towards privacy aware users, the site proposed users to test their browser uniqueness and measured it by examining the HTTP request for the following variables: User Agent, HTTP ACCEPT headers and if cookies were enabled; the following from JavaScript AJAX posts: screen resolution, time zone, browser plug-ins; and from Java or Flash applet: System fonts.

Table 1 – Source of variables

Variable	HTTP	JavaScript (AJAX)	Java/Flash Applet
User-Agent	X		
Cookies	X		
Screen resolution		X	
Time zone		X	
Browser Plugins		X	
System Fonts			X

Since the user has control over the variables being checked, for example, browser plug-ins, system fonts and screen resolutions. Those variables can be changed, which would result in an another “fingerprint” being generated for the browser. However, the research found with a non-optimized heuristic algorithm was able with success rate of 99,1% guess the correct original fingerprint (Eckersley 2010).

Was also noted the plug-ins/techniques which supposedly increase privacy may have the inverse result, due to the reduce number of people using them, examples are: User Agent Spoofing in which the User Agent even though is not real is not common enough and Flash blocking where the Flash appears in the list of plug-ins but the browser is not able to execute Flash methods which implies a browser with Flash blocked.

JavaScript Performance Fingerprinting

This methodology was presented in W2SP 2011: WEB 2.0 SECURITY AND PRIVACY 2011 by By Keaton Mowery, Dillon Bogenreif, Scott Yilek, and Hovav Shacham the base of their method is that most popular web browsers are constantly looking for ways to improve to increase their market share. These enhancements are particular to each browser, which allows to uniquely identify the browser by the performance of specific types of JavaScript code.

The performance of the JavaScript is analyzed in benchmark utilities, this study used SunSpider and V8 JavaScript and measured how much time each type of code took to execute. With this, was possible to create a simple method to compare the benchmark with a known version of the browser, because there were significant differences in the JavaScript performance profile of each browser (Mowery *et al.* 2011).

For the Operating System detection using only performance measurements is difficult because of the slim effects that the OS has in the performance of the JavaScript.

The main variant is the JavaScript engine which is related with the browser. The team found, however, in a specific Firefox version in which was possible to recognize the OS based on the benchmarks. A similar structure, like the one used on the browser detection, was made to compare the benchmarks to a known system.

Regarding the architecture of the CPU detection, the team took a different approach with the 1-Nearest- Neighbor, this method works by looking for the closest match of X in the data sample, the more samples available the more precise it is. This process allows to avoid averaging due to the mismatch between the marketing name and the actual chip. The team found cases where the marketing name had different chips. This implied to consider a valid match each architecture had to have a minimum set of samples, this was the least precise of the three tests with a 45,3% of success.

This method of fingerprinting, even though was inefficient, taking more than 3 minutes due to data quality constraints, such as, other browsers processes and other JavaScript code executing,

but despite this shows it's possible to deduce the device hardware without using any JavaScript API's.

NoScript Whitelist Fingerprinting

This methodology was also presented in in W2SP 2011: WEB 2.0 SECURITY AND PRIVACY 2011 by the same team but focus on the existence of the plug-in NoScript in Firefox. This plug-in objective is to prevent the execution of unwanted JavaScript code without the specific authorization of the user, the default behavior is that every site is blocked and only frequently visited domains should be white-listed, which means, it will be possible to execute the JavaScript code from those domains.

Browsing enhancing technologies such as Responsive Design and interactive pages require JavaScript to work. This means the user white-list will contain a lot of private information regarding browsing habits (Mowery *et al.* 2011).

Since the JavaScript from non-white-listed domains won't be executed, if there is a page able to get a remote reference to a JavaScript file from another domain, will be possible to know if the domain is white-listed, this is viable because only a few websites require authentication to remotely access their JavaScript content.

After accessing the remote JavaScript file, it is easy to check if the domain is white-listed because even if the execution gives error, means it was executed, which translates to the site being white-listed. However, if the script is not able to check for any object created by the remote JavaScript means it wasn't executed, hence, not white-listed.

To get a sample of valid sites to check the team first gathered a list of valid domains from the Alexa Internet, a company which provides the most visited sites, being the two most important rules for the site validity: having JavaScript content and the JavaScript being hosted in the same domain as the website.

To test this concept, an Iframe, which represents a nested browsing context (embedding another HTML page into the current page) with the remote JavaScript code to execute was created for each domain, after validating if the test was successful or not the Iframe destroyed itself, measures to not overuse the resources were implemented by limiting the number of simultaneous active Iframes.

This method assumes the user gave permission to the domain, which hosts the previously mentioned Iframes, to execute JavaScript. The performance of this method varies depending how many white-listed domains are set in NoScript the more domains are allowed the more time it takes. From Alexa Top 1000, the team considered 689 met the search parameters, in an ideal situation, NoScript disabled which means every site will be checked the process took approximately 2 minutes, no attempts were made to optimize the process.

This study shows that even a method of protecting the user, could be used to create a profile of his online life by allowing a 'rogue' site the execution of JavaScript only once.

Canvas Fingerprinting

Canvas fingerprinting was first mentioned in a research by Keaton Mowery and Hovav Shacham from the University of California, they propose a method using HTML5 element canvas and the WebGL, a JavaScript API used for rendering 3D and 2D graphics. The concept behind this fingerprinting methodology is an execution of code which renders text and WebGL scenes to a canvas page element, then with their native methods examine the pixels produced.

HTML5 new page element, canvas, is essential for this process because it provides an area where it can be programmatically drawn on and is a standard, which means it is available in most popular browsers. After the drawing process, it allows for pixel extraction with a method that returns a data URL consisting of Base64 encoding of the entire content of the canvas.

Since performance is essential for most of the normal uses of Canvas, inputs from the web page are passed deeper into the software stack which links the browser, the operating system and hardware. This translates to websites having access to more recourses which increases the performance but it also leads to variations on the behavior of the browser depending on the available resources.

Since canvas element provides pixel URL-encoded images, the team used a simple process to compare if two images were the same. A pixel-level difference is obtainable by comparing two images, an obtained image from the website and one from a defined set of images. Then subtracting the pixel colors, the expected result is a black rectangle meaning that every pixel is identical, RGB (0,0,0) represents black.

The test has three parts having the first two an additional step. The first is named Arial Text and uses the font Arial font which has been released over 30 years ago, it uses two different writing methods to write a specific string, each setting a different font size, in a 300 sample it was found, at least, 43 different set of pixels were drawn for each method.

The next step is similar to the Arial Text, but instead of using a local font, it downloads one from the internet using CSS3, Cascading Style Sheets, WebFonts feature, again the two methods were used, varying in the font size. From a 294 valid user samples resulted in 45 distinct ways to render the sentences.

Lastly the WebGL test, where in a canvas it's drawn a number of polygons with a simple texture and a light source, from the 270 valid samples there were 50 distinct renders of the scene.

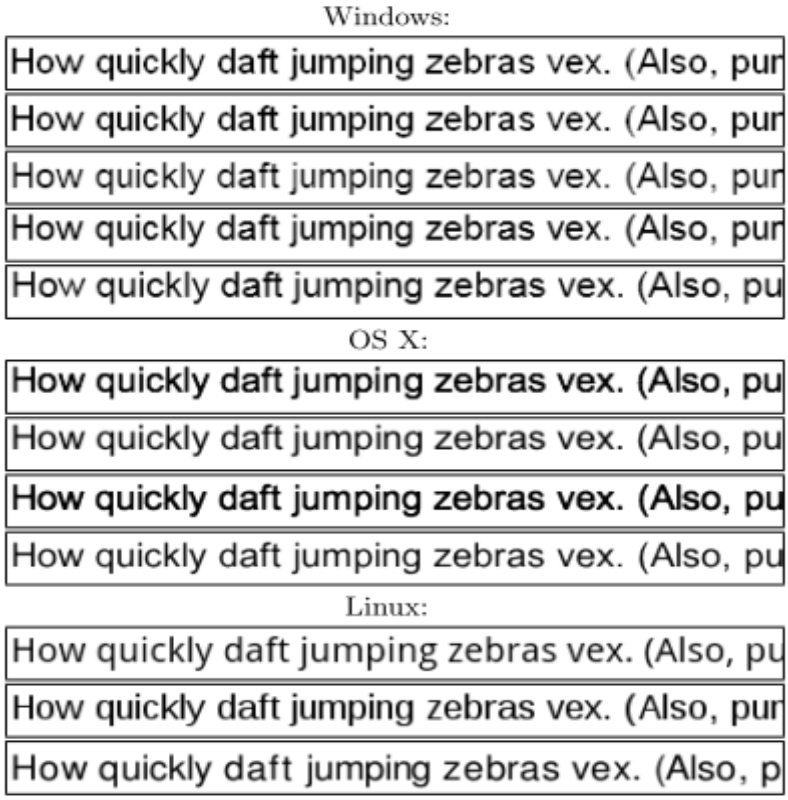

Figure 3 – Different ways to render "How quickly daft jumping zebras vex" - (Mowery & Shacham 2012)

Even with a very small data sample and imperfect method of exploring the uniqueness of each graphic system was possible to create a hardware profile of the devices which visited the website. It is also note worth, that system settings such as "Smooth edges of screen fonts" in Windows had visual impact on the image rendered, which demonstrates the effects of the user settings in the operating system has on the image drawn on the browser.

2.2 Tracking Prevention

This section details the main browser plug-ins available which offer internet browsing with the reduced risk of being tracked by blocking advertisement and some web bugs.

2.2.1 Ghostery

Ghostery is owned by Ghostery Inc. and is a browser plug-in available for all major web browsers and works by blocking third-party requests to domains considered to be trackers. It finds and disables cookies, scripts, and pixels used for tracking. It notifies users about which companies have been blocked and allows the option of selectively unblocking these companies.

Ghostery analyzes the requests made by a browser and compares them against a database of known trackers, this tool maintains a list of confirmed trackers and makes it available to everyone.

For each tracker on the list, it has a description, the URL of the tracker's company website and potentially the postal address, which could be relevant when identifying where the user information is being sent to.

The browser plugin divides its detected scripts into five categories (Castelluccia *et al.* 2013):

- Ad: advertisements provided by the ad-networks;
- Tracker: scripts which perform tracking (often using very sophisticated behavioral analysis);
- Analytics: utility scripts for Website creators allowing them to discover various statistical details about their visitors;
- Widget: small Web applications such as clocks, weather tables, and others. Other examples include Facebook Social Plugins, Google +1, etc.;
- Privacy: typically, a script disclosing privacy policies and practices related to ads.

The main positive points of Ghostery are the increase of the user awareness, because of the on-screen alerts of the trackers present on the web page being visited, and an easy installation. In contrast with the ease of the installation, the negative point is the default settings, which don't block any tracking, Figure 4.

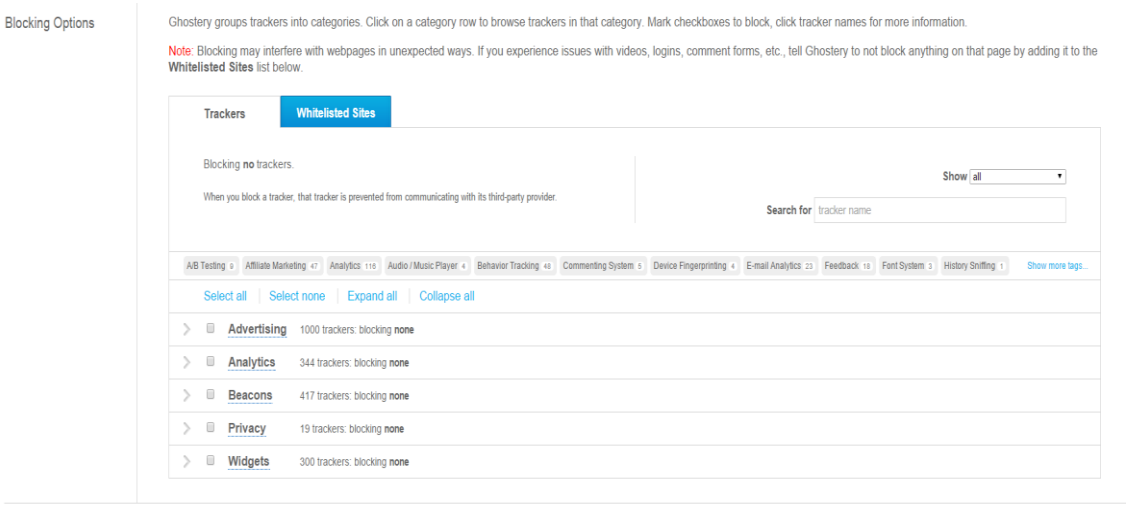

Figure 4 – Ghostery default blocking settings.

2.2.2 Adblock Plus

Adblock Plus is an open-source web plug-in, created by Wladimir Palant in 2006, which relies on a filter subscription, there are several subscriptions maintained by people not related directly with Adblock project. This tool blocks third-party resources, mostly unwanted advertisements and web bugs.

Wladimir Palant together with Till Faida created Eyeo which owns Adblock Plus and is responsible for the Acceptable Ads project, which defines a set of criteria that define an 'acceptable ad' using principles such as placement, distinction from the page content and size. Companies can apply to be part of the list, however, must pay, making this the main source of income for Eyeo and the Adblock project.

This tools works by matching URL patterns embedded in a web page against a set of blacklist patterns defined in the filter subscription (Liu *et al.* 2013).

Adblock unlike other tools uses a quieter approach, it simply blocks the tracking content of the web page giving the user an ad-free experience maintaining the correct structure of the page.

The main criticism it suffers is the lack of explanation regarding the different subscriptions, without the user going out of its way to search for the information.

2.2.3 Disconnect Me

This tool is an Open Source project started by a former Google employee with the aim of giving back the users the control of their browsing history (Fiegerman 2013).

This project began with blocking Facebook from continuously gathering information about everything it was being done online even outside the Facebook domain. Started with a simple code, Code 1, which would block HTTP requests outside of Facebook domain to Facebook.

It grown and was made into an open source plugin that blocks not only Facebook but also the other trackers. During the navigation, the user can see which trackers are present in each page and decide to block or allow them through the interface.

It grown and was made into an open source plugin that blocks not only Facebook but also the other trackers. During the navigation, the user can see which trackers are present in each page and decide to block or allow them through the interface.

```
/*The domain names Facebook phones hone with lowercase. */
const DOMAINS = ['facebook.com', 'facebook.net', 'fbcdn.net'];

/*
    Determines whether any of a bucket of domains is part of a URL, regex free.
*/
function isMatching(url, domains){
    const DOMAIN_COUNT = domains.length;
    for (var i = 0; i < DOMAIN_COUNT; i++)
        if (url.toLowerCase().indexOf(domains[i], 7) >= 7) returns true;
        /*A valid URL has seven-plus characters ("http://"), then the domain.*/
}

/* Traps and selectively cancels a request. */
if (!isMatching(location.href, DOMAINS)) {
    document.addEventListener('beforeload',     function(event) {
    if (isMatching(event.url, DOMAINS)) event.preventDefault()}, true);
}
```

Code 1 - Original Disconnect me code

2.3 Tracking Measurements

Web privacy, which was discussed before, is a topic which has been getting more attention by the media, but calculating web privacy is next to impossible. Despite several studies have been made in last years to help understand how the web privacy has being taken away from the online users.

WPM studies typically aim to attribute causality (i.e., to establish claims such as “the use of privacy feature X results in a 20% decrease in targeted advertisements”). Yet, the web is a complex, dynamic, interacting system with a multitude of actors. This introduces an incredible array of confounds, making such inferences very problematic to tease out (Englehardt *et al.* 2014).

2.3.1 Anti-Tracking Tool

In the market, there are several tools which claim to block tracking, like the ones mentioned in 2.2 topic. Some of those tools are open-source which means they could be used as a starting point for a development of a tool capable of identifying tracking mechanisms and the identities behind them.

Adblock Plus has a feature which is likely to pervert the results, which is the ‘acceptable ad’ project, which is the main source of income of the company behind this project, being such a relevant source of income it’s possible it might have underlying conditions which aren’t clearly stated on the tool site. Due to the uncertainty, this tool was excluded from the list of possible tools to be used on this thesis.

The other open source tool is Disconnect-me, Figure 5, this tool has the advantage over Adblock Plus due to how it works, instead of just removing the advertising from the page it provides, for

each page, a list of third-parties detected and their affiliation, for example if they are related with social plug-ins, advertising or analytics.

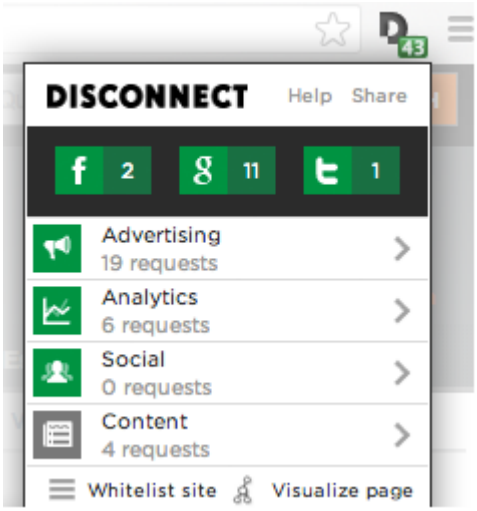

Figure 5 – Disconnect-me interface

The mentioned functionality could be altered to log the results in a file or a database, and after it would be possible to analyze the results. For this methodology to work, a script capable of sending a list of addresses to the browser would have to develop.

The extent of the capabilities of detecting tracking mechanisms is unknown so there is a risk similar to Adblock Plus of overlooking tracking mechanisms.

2.3.2 FourthParty

FourthParty is a dynamic web measurement platform and was developed by Jonathan Mayer and John Mitchell from Stanford University in 2012. This tool was designed with three core principles:

- General-purpose instrumentation, a detailed control which logs the result only once and removes the need for developing purpose-built tools;
- Production Web browser, the tool operates from a common browser which permits the use of browser add-ons for automation;
- Standardize log format, which provides an easy to compute result.

In order to fulfill all the requirements detailed, the team choose to develop a Mozilla Firefox extension, using the browser API's to control the HTTP traffic, DOM windows, cookies and resources being loaded. Fourth-Party also instruments JavaScript API calls on the window, navigator, and screen objects using getters, setters, and ECMAScript proxies (Mayer & Mitchell 2012). The results of the analysis are stored in SQLite database.

Being a browser extension, in long crawls the computational requirements for each browser instance has to be taken in consideration as it can put a strain in the available resources, which reduces the parallelism possibilities in less robust systems.

2.3.3 OpenWPM

OpenWPM is a web privacy measurement framework which simplifies the method of collecting data for privacy studies on a scale of thousands to millions of websites. OpenWPM is built on top of Firefox, with automation provided by Selenium web driver. It includes several methods for data collection, including a proxy, a Firefox extension, and access to Flash cookies.

It's a python framework which provides high scalability, crash recovery and reproducibility to web crawls with the goal of identifying third-parties present in the websites scanned, it's maintained by Steven Englehardt from Princeton University and the code is open source available in GitHub (github.com/citp/OpenWPM/).

OpenWPM is a flexible, stable, scalable and a general web measurement platform, this solution fills the infrastructure vacuum on web privacy measurements.

3 Solution

OpenWPM, where WPM stands for Web Privacy Measurement, already has six completed studies on the subject of web privacy and several others in progress across several institutions plus a partnership with Electronic Frontier Foundation (Englehardt *et al.* 2015). This open-source tool was designed to provide a high level of reproducibility, which means it will be possible to standardize privacy measurements.

Using, OpenWPM, an existing framework for the research of Portuguese market, has numerous advantages, several implementation challenges were already overcome, the finished researches show the framework works and provides accurate and reliable results and, most of all, will allow this framework to evolve with the possible improvements that this research will require. This, in the long term, will make the framework a reference for other privacy measurement studies.

In this chapter, it will be provided a detail analysis of the framework and the method used to parse the results obtained.

3.1 OpenWPM

Open Web Privacy Measurement appeared out of the need to standardize the web measurements regarding privacy, lots of studies in past years have been made but each used their own framework which meant the studies could not be accurately compared to each other's. This project was developed by Steven Englehardt, Chris Eubank, Peter Zimmerman, Dillon Reisman and Arvind Narayanan from Princeton University.

The team analyzed the challenges and features of 30 previous studies to designed and developed an open source framework which could serve not only for academic researchers but also regulators and press, since online privacy is becoming an increasingly hot topic.

OpenWPM is built on Python and is divided in three modules: browser manager, task manager and data aggregator, Figure 6. The task manager monitors the browser managers, which is responsible for converting high-level commands, for example clicking a link or scrolling the page, into automated browser actions. The data aggregator receives and pre-processes data from instrumentation for analysis.

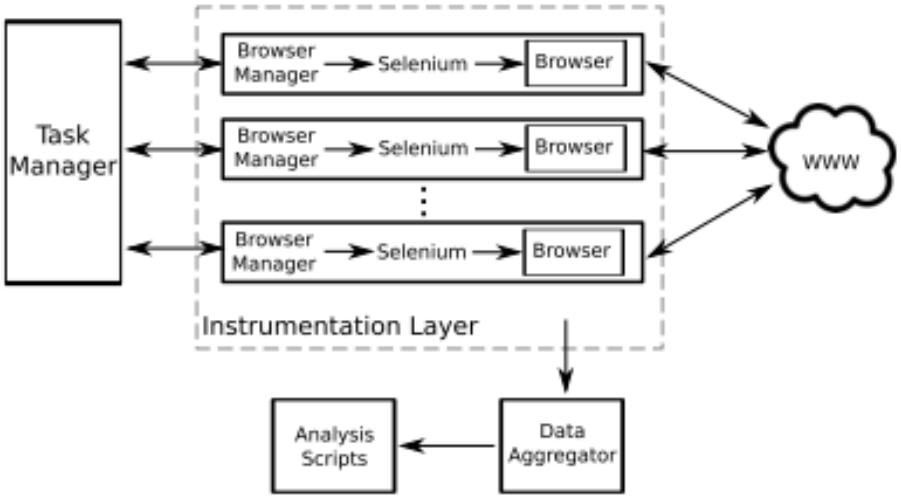

Figure 6 – OpenWPM overview (Englehardt *et al.* 2015)

3.1.1 Browser Manager

This tools uses Selenium which is prone to frequent crashes and freezes, this requires Selenium to be abstracted from the other components to avoid its malfunction to corrupt the data from other modules, in case of failure, is capable of returning to the state it was before the crash. Additionally, all code related with Selenium is limited to this module.

Each browser manager represents a browser and indirectly a user, which means each instance can have specific configurations, which has several benefits, for example, testing the impact of changing certain settings or to help simulating different users.

It is responsible to convert high-level commands, such as, clicking a link or scrolling the page into Selenium commands. However, in large scale studies where it's needed to crawl thousands of websites the resources must be taken into consideration, as such, browser manager uses headless containers, which operates in the same way as a normal browser but without a graphical interface. On the other hand, this possess a problem, the content on the web page in some studies, regarding privacy, is relevant.

For these cases in which the visual content is of interest, the manager uses Xvfb, virtual frame-buffer server, which works by emulating a frame-buffer using virtual memory and

PyVirtualDisplay a python wrapper for Xvfb. This allows for print screens of the content of the web page to be taken in a headless manner saving resources and allowing greater parallelization.

3.1.2 Instrumentation

Instrumentation is a process which runs alongside the browser manager and is responsible for measuring and logging the connections each instance of the browser manager is making to the web, for this Mitmproxy was used.

Mitmproxy is an open source proxy that allows intercepting HTTP and HTTPS connections between any browser and server using a typical man-in-the-middle attack.

It accepts connections from clients and forwards them to the destination server. The goal of mitmproxy is to let an attacker monitor, capture and alter these connections in real-time(Heckel 2013). Being able to capture every connection being made allows to monitor where the requests are being made to.

3.1.3 Task Manager

The task manager provides an interface to control multiple browsers at the same time, while abstracting error handling by restarting the browser manager when it detects it has crashed or didn't complete the command within a certain time limit, which could indicate a freeze happened and the outcome is the manager isn't going to complete its commands.

By assigning each browser manager a thread with a set lifetime, it can easily recover from the browser crashes by automatically copying its state to a new location, killing the remaining processes associated with the frozen browser and starting a new browser, with the state previously saved in a temporary location.

It allows up to four ways to sending out commands to the browser managers: First-come, first-served on which whenever a browser manager becomes available it executes the respective command; individual browser; all synchronously; all asynchronously.

3.1.4 Data Aggregator

This process is responsible to gather and serialize the information collected from the other modules and after doing simple manipulations to homogenize the data it stores it in database, the default installation is an SQLite database due to its lightweight. However, it can be swapped for another SQL engine.

Since the aggregator must be reachable from every browser manager and the instrumentation processes, it uses a socket interface. The socket interface increases the protection against

corrupt data from crashes in other modules and if needed also allows SQLite to be replaced by other database framework, due to the abstraction extra provided by the socket interface.

3.1.5 Database

The data aggregator process stores the information gathered from browser and task manager, in a structure, Figure 7, this is independent of the database engine and stores all the information needed to perform a future data analysis.

Using SQLite has significant advantages when using in low resources system, because is the only database which allows creating a database server-less with minimal resources. This, however, has a drawback, the database engine is limited in the natively supported features. To overcome this most of the analysis must be supported by the framework or an external tool.

During the evolution of OpenWPM, the table Site_Visits was introduced, which simplified the identification of which website a certain cookie belonged to, but made so that there is repeated information within the foreign keys. With a Visit_ID is possible to uniquely identify the Crawl_ID, but the column is also present in most of the tables. This shows the constant evolution of the framework is having and is still far from a final version.

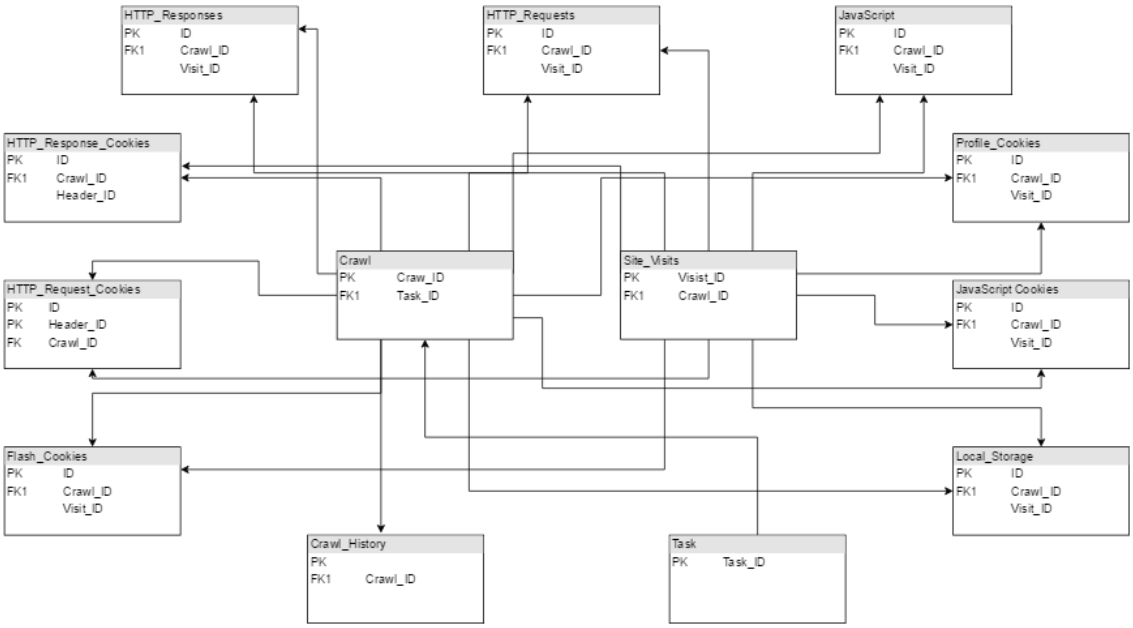

Figure 7 – Simplified Database Diagram

Since every object is linked to a specific crawl and the site visited, the data can be easily compared between crawls to detect changes, for example, additional cookies or another third-party by presence of an additional HTTP request.

The information stored in all current known storage vectors are saved on this structure which allows easy access to the locally stored information and to know to which domain it belongs to,

although, it brings up a potential issue, if trackers start storing additional info in another vector this will hide the results from the data gathering process.

3.1.6 Machine

This framework was designed to operate in Linux, more specifically Ubuntu 14. Due to the nature of the test it is recommended to run it in an independent server to avoid IP related problems, such as the IP being marked as a bot which could alter the behavior of the websites visited.

Additionally, the crawl will consume a lot of hardware resources for a long period of time due to the number of sites needed to complete the study.

A server allows for the test to be replicated as many times as deem necessary without worrying about blocking a personal computer for long periods of time and the tests can be done from different locations, this could be used to test if the location of the user has impact on the results.

3.2 Analysis tool

Due to the amount of data the framework produces, it will be necessary a tool capable of extracting the thousands of cookies created in the crawls and registered in a SQLite database. SQLite database is a C library that provides a lightweight disk-based database that doesn't require a separate server process and allows accessing the database using a nonstandard variant of the SQL query language (Python Software Foundation 2016). To achieve this, many standard native functions which are present in other database software's were striped, this implied the analysis couldn't be done at database level directly.

Three options emerged, extending the post processing capabilities of the framework using python, developing a local Java application that would connect to the output of the crawl and do the parsing of the results or changing the output from SQLite to MySQL.

Extending the framework has as main advantage the contribution to the open source development of the framework and using a single tool to obtain the data and provide the analysis of it. However, it also has big disadvantages such as, it being a post-processing step implies the analysis will only be executed after running the crawl, for small crawls this isn't a problem since each run would take an hour. But for big crawls they can easily take 8 hours or more to complete. And with the current architecture is not possible to just execute the analysis, which emphasizes the constraint of the time needed to execute the crawl.

The change from SQLite to MySQL would require changes to data aggregation module, because even though it was designed to be capable of easily change the database, as explained in 3.1.4, where it logs all the information which is being executed in the browsers of the crawl, there are

still areas of the code not abstract enough to change the database, in specific the Firefox extension responsible for logging the profile cookies.

The alternative, would be to convert the SQLite output to a MySQL database after the crawl was completed, but even this has no guarantee to provide all the necessary tools for the analysis. As such this hypothesis was set aside.

The last option, a local Java application. This has in its favor the possibility of once obtained the database file it can be executed as many times as necessary. Additionally, it allows for quick bug fixes and even the addition of additional functionalities, without having the execute the crawl again. The disadvantage of this alternative is very small: the database file must be moved from the server to the machine with the Java application.

As explained, the advantages of the local Java application are greater than the expansion of the post-procession expansion of OpenWPM, so for those reasons the Java application was selected to do the post processing of the data gathered.

3.2.1 Implementation

The Java application implemented is very simple, it connects trough JDBC to a local SQLite database, obtained from the server which executed the crawl.

Expanding on the explanation in 1.1, a first party cookie is a cookie created by the website which the user is visiting, those are identified by sharing the same host and the website being visited. These cookies are not exempted from being a source of tracking but each time the user navigates to a website it's shown a message with the ToS, like the fowling (from www.paypal.com) "By using PayPal.com you agree to our use of cookies to enhance your experience".

Unlike the previous, in third-party cookies the host doesn't match the website being visited and most of the times the same host appears in multiple websites, furthermore, they may also share the same Value field, which is indicative of tracking.

To separate the first party cookies from third-party cookies it was used a matching algorithm to find, in terms of percentage, how identical the host of the cookie is compared to the website. As is shown in Table 2, first party cookies can take many formats and to reduce false positives as much as possible the algorithm which determines if it's a first party or not, checks if the website and the host strings are 70% similar, to increase the accuracy the websites strings are striped from "www" and "http".

Nevertheless, despite the efforts there are still cases which the algorithm cannot identify a first party cookie, like the example shown in Table 2, where the host is an IP address and the website is "abola.pt". The IP address redirects to the website, however, the algorithm is not capable of detecting these cases.

Table 2 – First and third-party Cookies example

Website	Host	First Party	Third-party
sapo.pt	pub.sapo.pt	X	
sapo.pt	.sapo.pt	X	
sapo.pt	.smartadserver.com		X
abola.pt	www.abola.pt	X	
abola.pt	193.126.232.45	X	
abola.pt	.ad-srv.net		X

4 Method

In this Chapter, it is defined the main steps needed to execute the research from the gathering of the sites, setup up the framework to the goals regarding demonstration of the results obtained.

Regarding the sites, there is an analysis of the most visited websites in Portugal, as well as, the distribution of the top-level domains.

Lastly, it also details the process to validate the results of the framework.

4.1 Target Websites

Portuguese access to internet is around 30% with over 50% with regular use (ANACOM 2015), which means it is full of potential targets to third-parties.

OpenWPM requires a list of sites to visit and analyze, the focus of this study is the Portuguese market, to be able to get a decent data sample the ideal method is using an API from a service which can provide the most visited websites.

Alexa is a company which provides for free a list with the one million most visited sites worldwide, but through a paid API it provides the top X most visited websites of the selected market.

With a way to get the most visited sites in the Portuguese market, the next goal is to define the size of the sample, too little the result won't be relevant and too big will get too much noise from smaller sites which are unlikely to have third-party codes embedded because they are not profitable for tracking.

The number decided is the most visited 700 websites, which means every popular web page will be checked but also medium sites while avoiding most the smaller ones, which represent the majority of available sites even though they are not often visited. The company provides a Java tool which connects to an API to obtain the requested sites, however, the API has a limitation of only returning 99 sites by request. The code provided was changed, to loop 7 times and store both the XML returned and a text document with a list of domains.

4.1.1 Websites Analysis

The crawls consist of the analysis of the 700 more popular websites visited by a Portuguese IP address, this includes tourists and emigrants. The presence of both is small compared with native Portuguese users, but their presence in a large data sample such as this is notable.

In Table 3, is shown all the top-level domains, the 'others' include all the top-level domains that appear only once. Even though a website doesn't have '.pt' as a top-level domain doesn't exclude it for being a Portuguese website, the opposite is also true a website with '.pt' is not certain it's a Portuguese website. But latter is very uncommon, a site with the top-level domain '.pt' not being a Portuguese.

Table 3 – Top-Level domain distribution

Top Level Domain	Count	Percentage
.com	352	50,29 %
.pt	196	28,00 %
.net	36	5,14 %
.br	25	3,57 %
others	17	2,43 %
.org	16	2,29 %
.tv	8	1,14 %
.uk	7	1,00 %
.de	6	0,86 %
.me	5	0,71 %
.io	5	0,71 %
.es	4	0,57 %
.ru	4	0,57 %
.eu	3	0,43 %
.lub	3	0,43 %
.fr	3	0,43 %
.co	2	0,29 %
.se	2	0,29 %
.xyz	2	0,29 %
.cc	2	0,29 %
.to	2	0,29 %

Moving from the top-level domain to the actual content of the website, each site of the 700 was manually verified and classified according to its content, Table 4. This table shows an overview of the diversity of content present in the websites which will be subject to several crawls to identify the third-parties present.

This provides an overview of the type of content the Portuguese market actively searches for in its daily online lives.

Table 4 – Websites category distribution

Category	Number	Percentage
Internet and Telecom > File Sharing	61	8,71 %
News and Media	56	8,00 %
Business and Industry	53	7,57 %
Internet and Telecom > Social Network	45	6,43 %
Internet and Telecom > Ad Network	35	5,00 %
Internet and Telecom > Search Engine	29	4,14 %
N/A	29	4,14 %
Adult	28	4,00 %
Finance > Financial Management	27	3,86 %
News and Media > Magazines and E-Zines	26	3,71 %
Shopping > General Merchandise	25	3,57 %
Law and Government > Government	22	3,14 %
Arts and Entertainment > TV and Video	20	2,86 %
Computer and Electronics > Software	20	2,86 %
Arts and Entertainment > Humor	20	2,86 %
Travel > Accommodation and Hotels	19	2,71 %
Career and Education > Universities and Colleges	17	2,43 %
Games > Online	15	2,14 %
Shopping > Clothes	15	2,14 %
Reference > Dictionaries and Encyclopedias	14	2,00 %
Shopping > Classifieds	11	1,57 %
Internet and Telecom	10	1,43 %
Sports > Soccer	10	1,43 %
Internet and Telecom > Web Hosting	10	1,43 %
Shopping > Consumer Electronics	10	1,43 %
Career and Education > Jobs and Employment	9	1,29 %
Games	8	1,14 %
Gambling > Lottery	8	1,14 %
Internet and Telecom > Email	7	1,00 %
Games > Video Games	7	1,00 %
Sports	7	1,00 %
Business and Industry > Real Estate	6	0,86 %
People and Society > Relationships and Dating	6	0,86 %
Shopping	4	0,57 %
Arts and Entertainment	3	0,43 %
Internet and Telecom > Telecommunications	3	0,43 %
Arts and Entertainment > Movies	2	0,29 %
Autos and Vehicles	2	0,29 %

4.1.2 Crawls

To obtain a clear picture to the amount of third-parties the user is exposed in regular basis, the crawls were done in server hosted in Amazon Frankfurt AWS data center, unfortunately, no Portuguese based datacenter offered the conditions to easily create and destroy VPS's while providing a different external IP. The different external IP is an additional layer to prevent sites visited in different crawls identify them as the same user, explained in 3.1.6.

For the first crawl, the configuration used was the most basic possible, a browser moving from web page to web page without any tracking prevent measures active. Nevertheless, the framework uses some configurations, detailed in 4.2.1, to avoid flagging the browser as a bot, so those configurations were kept.

The following crawl, was designed to analyze how the extension Adblock Plus, detailed in 2.2.2, fared in preventing tracking from taking place. This tool was chosen because is one of the most popular extensions, with almost 21 million users in Firefox, regarding privacy more specifically to block unwanted advertising.

The third crawl aimed to test the extension Ghostery, 2.2.1, an anti-tracking tool in contrast to ADP which is an ad blocking tool, and compare the results obtained with Adblock Plus against this extension, to validate which was, overall, more effective.

4.2 Analysis

It was used OpenWPM as the framework to collect information. In this section, it will be provided some details about OpenWPM configuration, execution and the goal with the extraction of the results.

4.2.1 Automating Data Collection

For this privacy measurement, it is essential to have websites regarding the browser of the crawl as a real user, in contrast to, a bot, OpenWPM provides ways to achieve this by simulating random clicking on the page, scrolling and delaying navigating between pages. But this is not enough; it is necessary to have a way to make each browser as unique as possible. The way to achieve this is by defining different browser settings, this will have the additional benefit of when running the crawl again for the same website, new configurations could be selected and validate their impact in the new results obtained.

When running multiple measurements based from the same IP to validate the research there is a high risk of the site marking the IP has a bot, therefore, invalidating or adulterating the results. To avoid this, there is a limit of possible configurations that have enough significance to be considered a unique in a fingerprint process, and an additional step is needed, which is a user profile. The need for a user profile comes from the premise the user is being tracked, and third-

parties have a profile of the user from previously visited websites. With this premise, before starting the measurement crawl, a profile can be simulated. There are two ways of doing this: (i) using real user information which can be obtained by crowd-sourcing or (ii) generate a profile.

The crawling capabilities of OpenWPM can be used to generate the user profile mentioned above. This can be done by defining a list of websites, which will represent the virtual user preferences, then simulate a crawl to those websites, and save the cookies created in the process. With a profile and somewhat different browser configuration, it is possible to simulate several users in the same machine. Although, due to constant timeout errors obtained in previous test runs, the timeout setting was greatly increased which required more resources and limited to a single instance of OpenWPM per server.

4.2.2 Extracting Information

With the results obtained from OpenWPM it will be possible to generate charts based on the frequency tracking mechanisms are detected on a web page and how popular the website is. Furthermore, details regarding how many third-parties are present based on the page rank and which are the most predominant trackers on the web.

The analyzes of the tracking mechanisms on the Portuguese market will be important in understanding how exposed the Portuguese user is to such methods and, based on the third-party prevalence, how capable are they to create and maintain user profiles.

The overview of the most common trackers on the web will allow users to be aware of what features from the common browsers and related technologies they exploit to gather information. This could inform users of what is happening on the Portuguese web or help companies that want to provide a service of protecting users against tracking.

Web pages, normally, have more than one tracking mechanism embedded in their code, for example, the three major social networks with their social plug-ins are available in most web pages, that is a small example of the possible number of trackers in a web page, with the analysis of how many third-parties are presented it will be possible to draw conclusions regarding where the user private information is being sent to.

4.3 Java Tool

In this section, it will be detailed the simple Java tool used for the analysis of the results obtained from the framework OpenWPM, described in detail in section 3.1.

It will demonstrate the challenges faced and the solutions found to overcome them and the several data structures obtained to fulfill the objectives of this research.

4.3.1 Base Tool

The java tool provides a connection to the database of the crawl, a list of cookies with their website, host, name and values and a list of the unique third-parties. These data structures will be the base of the analysis which will be detailed in the following chapter.

On the database, to simplify the query used to gather the dataset, a dynamic view was created to collect in a single result set, all three types of cookies collected associated with the site on which they were stored, Code 2, additionally it was limited to only return cookies in which the value was 5 or more characters to avoid results with only flags in its values.

```
SElECT sv.site_url as siteUrl, c.rdParty as host, c.type,
       name, c.value, c.httpOnly, c.isSecure
FROM (
    SElECT visit_id, crawl_id, Host As rdParty, 'javascript_cookies' As type,
           name, value,is_http_only as httpOnly, is_secure As isSecure
    FROM javascript_cookies
    UNION ALL
    SElECT visit_id, crawl_id, Host As rdParty, 'profile_cookies' As type,
           name, value, isHttpOnly as httpOnly, isSecure As isSecure
    FROM profile_cookies
    UNION ALL
    SElECT visit_id, crawl_id, domain as rdParty, 'flash_cookies' As type,
           key As name, content As Value, -1 as httpOnly, -1 AS isSecure
    FROM flash_cookies
) c, site_visits sv
WHERE 1=1
AND c.visit_id=sv.visit_id
AND c.crawl_id=sv.crawl_id
AND length(c.value) > 6;
```

Code 2 – View to obtain all the cookies

Based on the result set above, the first step is to obtain a collection of the third-parties, the most important part is to guarantee that there are no repeated third-parties. To avoid having to implement a logic to scan a list before inserting a new Object, the Java Collection Set was used.

A Set is a Java Collection that cannot contain duplicate elements, it models the mathematical set abstraction (Oracle 2015). Using the implementation HashSet, which stores its elements in a hash table, is the best-performing implementation, however, it makes no guarantees concerning the order of iteration. Since the objective is to have no repeated third-parties and the best possible performance, due to the dataset containing several thousands of cookies registered and order not being important factor, made this implementation the best option.

4.3.2 Proliferation of Third Parties

One of the main goals is to identify which are the main trackers on the Portuguese market, to do so, is necessary to check the third-party cookies and identify which are the entities behind them. This is not possible, since entity is not a field of the cookie, the closest obtainable is the information regarding host of cookie.

With the host, it will be possible to measure the reach an individual tracker has over the online behavior of a user, and will also serve another purpose by validating the impact that tracking preventions tools have in the number of third-parties present in the overall execution.

The base tool provides the list of third-party cookies and a list of unique third-parties. With those two collections, it's possible to get a list of websites which contain cookies from those third-parties.

However, due to usage of the Collection HashSet for the listing of third-parties had an unexpected impact. The method Add of HashSet uses the method equals, inherited by the Object class, to verify if the object being added is unique.

The method equals from the class Object compares if two objects are the same, this is problematic when using a Set collection of objects which have a list of items that is constantly updating. In the case of this analysis, the object Third-party has a list of websites where the entity can be found, and is constantly being updated.

To solve that short come, the method equals of the class third-party was overwritten to only compare the unchangeable fields, this way the list of visited sites could be updated and added again to the listing, HashSet implementation still guarantees the content uniqueness.

4.3.3 Analysis of Websites

A very similar method was implemented to identify the reverse, that is, starting on the websites obtain a list of unique third-parties present on each website. As explained in the previous section, the key element is using the Collection Set which, already guarantees that an object being added is unique.

So, for each cookie the website is obtained and the third-party is added to list third-parties, then the website is added to the data structure, which according to the HashSet implementation when adding an already existing element it replaces it. For this to happen, as explained in the previous section, the method Equals was overwritten to match only inalterable fields.

4.3.4 Cookies Uniqueness

As it was pointed out before a cookie is a triple domain, key, value that can be used to track users. The tracking the cookies provide works by creating a cookie with the same key and value in several websites visited.

This provides way for the host of the cookie, the company behind the cookie, to be able to obtain information of the user online behavior by following the cookies inserted by them, since the key/value is unique to a specific device.

To identify those cookies is necessary is to start from the list of third-party cookies, detailed in 4.3.2, and obtain the cookies that share with at least one more cookie the same host, key, value. The next step, is to create a collection which gathers the information above, plus, a list of which websites contain the identical cookies.

The implementation of this analysis, starts by gathering cookies which share the same properties from the third-party cookie list, a list that contains only cookies in which the domain and the host don't match.

For each cookie, it is necessary to compare it against the initial list, which results in a very time consuming process. Within the nested loop, the first step is to exclude the cookie itself, otherwise every cookie would be part of the list because it shares the same key, value, host with another cookie on the nested loop, itself.

After excluding itself, it compares if the host and the key match. Although, for the value a simple comparison of strings is not enough, because random characters may be added to the field for unknown reasons. So, it was used a percentage matching algorithm to discover how identic the two strings are, similar to what was already used to compare host and domains to separate first party cookies from third-party cookies in 3.2.1, the matching percentage value defined was 70%. The result of this was a collection of third-parties that share the same host, the same name and the same value.

The following step is to organize the information, so a Set Collection was created and starting by the list previously created, each cookie was checked if the above conditions were met, if it already existed, the domain was added to the list of websites that contained the specific cookie.

The outcome, was a list of elements which contained the trio: host, key, value; And a list of websites which contained the cookies with those parameters.

4.3.5 Prominence

Prominence is a formula proposed in OpenWPM paper (Englehardt *et al.* 2015), instead of just taking into account the number of sites a third-party is present it goes further, it takes into consideration the rank of the website in which they can be found. This means a third-party

which appears only in 10 sites but are the top 10 has a higher value of a third-party that appears in the last 50 websites of the ranking.

The formula proposed is $Prominence(t) = \sum edge(s,t) = 1\frac{1}{rank(s)}$, where t is a third-party and s the site being visited.

The Java analysis tool, gathers the list of third-party, which contains a list of websites where each third-party is present. Followed by calculating the prominence, using the formula above. And lastly, sorting the list of third-parties by prominence.

4.4 Data Validation

Data Validation is an important step in any research, it validates if the results obtained are accurate. There are several ways to do this but in some cases, like the one from this research, validating the results in not possible because there isn't an algorithm which is able to demonstrate that the results are credible. In these cases, what it's possible to do is to validate the data source.

To validate the data source, there are two approaches that can be used use a different tool with the same sample and compare the results obtained with both tools, the alternative when there aren't other tools capable of doing the same experiment, is to run a smaller set of the source and validate if the results obtained are consistent.

OpenWPM is unique in the way it operates and as described in section 2.3, the closest tool available is FourthParty but this tool doesn't register Profile Cookies and several other JavaScript calls.

As such, the alternative left is to execute OpenWPM in its simplest configuration five times for a set of 50 websites. The results obtained are in, Table 5, where it shows the number of created cookies in those 5 executions, using the same constraints as defined in 4.3.1.

As the previous table shows, crawl 2 was the one with the least created cookies reaching a difference of 16,28% to crawl 5, the crawl with more cookies created. Comparing cookies with a length bigger than 6, which are the focus of the analysis, the difference is 15,91%.

Table 5 - Data validation comparison

	All	>6	Start Date
Crawl 1	3048	2489	2016-10-08 18:16:53
Crawl 2	2715	2214	2016-10-08 18:58:43
Crawl 3	3167	2577	2016-10-08 20:05:13
Crawl 4	3190	2593	2016-10-09 15:47:29
Crawl 5	3243	2633	2016-10-09 16:25:57

The deviation is bigger than expected, the causes for this discrepancy are impossible to pinpoint, because each crawl was executed in a different and automatically attributed server managed by Amazon the only control possible is the location of server and for every measurement used it was always used the same to minimize the impact of IP geolocation.

Although not ideal, if crawl 2 results were overlooked and focusing on the others crawls, the difference is 6,01% and 5,47% for every cookie and cookies with value bigger than 6, respectively.

The target goal was less than 10% difference between crawls. Even though, this number may seem high there are several valid reasons for the number of cookies created differentiate from crawl to crawl, for example, timeouts, temporary server unavailability or the heavy load on the local server. Those factors cannot be controlled or mitigated due to the duration of the crawl, estimated to last 8 hours for each crawl.

5 Results

In this chapter, it will be detailed the outcome of the measurements made with OpenWPM with different configurations. The configurations which were analyzed are identified in 4.1.2. Which are the following:

- No anti-tracking configuration
- Adblock Plus extension
- Ghostery extension

The extensions chosen were detailed in Chapter 2.2. Those configurations allow to compare how exposed an end user is to tracking without any protection, how much difference does having one of this tools make and to compare the efficiency of both tools.

The first measurement is the distribution of cookies created (5.1), followed by the reach of the third-parties (5.2), the ranking of third-parties (5.3), the identification of websites with more third-parties (5.4) and, lastly, details the exposure to tracking (5.5).

5.1 Cookie Distribution

The crawl of the 700 more visited sites by the Portuguese users without any setting to block or limit the cookie creation resulted in 48.876 cookies created.

In Table 6, it is possible to see the distribution of the cookies created by type during this crawl. This numbers reflects how much information is being stored client side. The cookies created belong to 2.103 different hosts, hosts can be first and third-party.

Table 6 – Crawl Cookie Type Distribution

Cookie Type	Number	Percentage
Flash Cookies	35	0,07%
JavaScript Cookies	20.495	41,93%
Profile Cookies	28.346	58,00%

Using the Java tool described in 3.2.1, the first measurement has the goal to identify how many of this subset of cookies are first and third-party, Table 7, demonstrates that 78,68% of the cookies created don't belong to the website being visited.

Table 7 – Crawl Third-party distribution

Type	Number
First Party	7.881
Third-party	29.081
Total	36.962

On the crawl with the extension Adblock Plus, which was detailed in 2.2.2, resulted in 19.868 cookies created.

The distribution of the three types of cookies created is detailed in Table 8. This, considers both first party cookies, created from the site being visited, and third-party, cookies created from other entities besides the one being visited.

Table 8 – Adblock Crawl cookie type distribution

Cookie Type	Number	Percentage
Flash Cookies	34	0,17%
JavaScript Cookies	11.299	56,87%
Profile Cookies	8.535	42.96%

The next step, is from the 15.266 cookies created, taking into account the cookies set a side which were too small contain significant information, separate and measure the first party cookies from the third-party cookies and compare the values to the previous execution.

Table 9, shows the distribution of third-party cookies in this crawl with Adblock Plus enabled, it is possible to see 47,36% of the cookies created don't belong to the website being visited.

Table 9 – Adblock Crawl Third-party distribution

Type	Number
First Party	7.385
Third-party	7.230
Total	15.266

The third crawl replaces Adblock Plus by Ghostery extension and the same tests were made, which resulted in a reduction of cookies created to 7.812.

The next table, Table 10, will contain the distribution of the type of cookies: Flash, JavaScript and Profile cookies.

Table 10 – Ghostery Crawl Cookie Type Distribution

Cookie Type	Number	Percentage
Flash Cookies	29	0,37%
JavaScript Cookies	5.268	67,43%
Profile Cookies	2.515	32,19%

With the Java tool, next is the separation of the cookies, Table 11. The percentage of third-party cookies in this crawl is 35,48%.

Table 11 – Ghostery Crawl Third-party distribution

Type	Number
First Party	3.773
Third-party	2.075
Total	5.848

Comparing the three crawls it's possible to start taking conclusions regarding the reduction of cookies being created.

Starting by the number of cookies, Adblock Plus reduced the number of cookies created 59,35% while Ghostery got the number down to 84,02%. This not only reduces the possibilities of tracking but also as the indirect effect of making the browsing quicker, for the end user, because the page doesn't have to load so many cookies.

Regarding the JavaScript cookies, Adblock Plus reduced to 44,87% the number of JavaScript cookies and Ghostery reduced 74,30%. Comparing profile cookies, Adblock Plus decreased the number to 69,89% and the other tool decreased it to 91,13%.

In regards to the separation of cookies into first party and third-party cookies Adblock Plus reduced the first party cookies created in 6,29% and third-party in 75,14%, on the other hand, Ghostery reduced the first party in 52,13% and third-party in 92,86%.

The reduction of third-party cookies created is surprising and a positive sign in the efficiency of this tools in reducing the tracking the user is exposed to.

Finally, regarding the difference flash cookies number in comparison with profile cookies and JavaScript cookies is somewhat contradictory with previous research dating from 2011 (Ayenson *et al.* 2011), but this was attributed to the fact Flash is currently being deprecated in favor of more recent technologies.

5.2 Third-Parties

With the list of third-parties obtained, the next step is to gather information on who are they. The crawl without any anti tracking tool generated 29.081 third-party cookies belonging to 916 different hosts.

In Table 12, is possible to see the 10 most predominant third-parties and the number of sites on which they can be found. Figure 8, shows the distribution of the number of third-parties by website.

Table 12 – Crawl Top 10 third-parties

Third-party	Number of Websites	Percentage
doubleclick.net	236	33,71%
google.com	206	29,43%
googlesyndication.com	140	20,00%
google.de	120	17,14%
adnxs.com	113	16,14%
scorecardresearch.com	108	15,43%
rubiconproject.com	91	13,00%
tpc.googlesyndication.com	89	12,71%
twitter.com	89	12,71%
criteo.com	75	10,71%

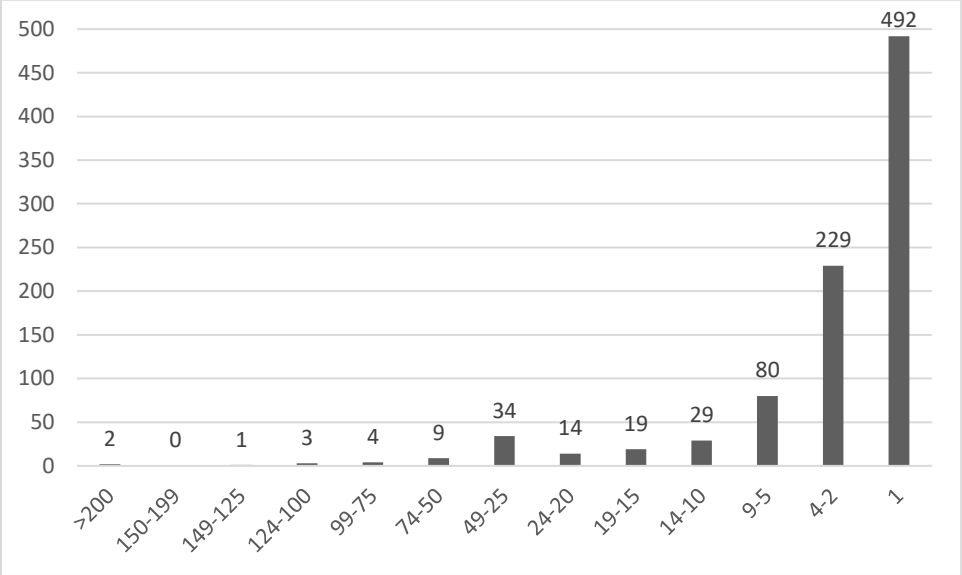

Figure 8 – Crawl Number of websites distribution

On the crawl with Adblock Plus extension it was created 7.230 third-party cookies belonging to 475 different entities. On the next table, Table 13, it will be possible to see the 10 most predominant third-parties from the ABP crawl. And the next figure, Figure 9, will display their distribution by websites.

Table 13 – Adblock Crawl Top 10 third-parties

Third-party	Number of Websites	Percentage
doubleclick.net	168	24,00%
scorecardresearch.com	96	13,71%
google.com	96	13,71%
twitter.com	71	10,14%
quantserve.com	57	8,14%
accounts.google.com	48	6,86%
taboola.com	43	6,14%
criteo.com	41	5,86%
cloudflare.com	40	5,71%
optimizely.com	38	5,43%

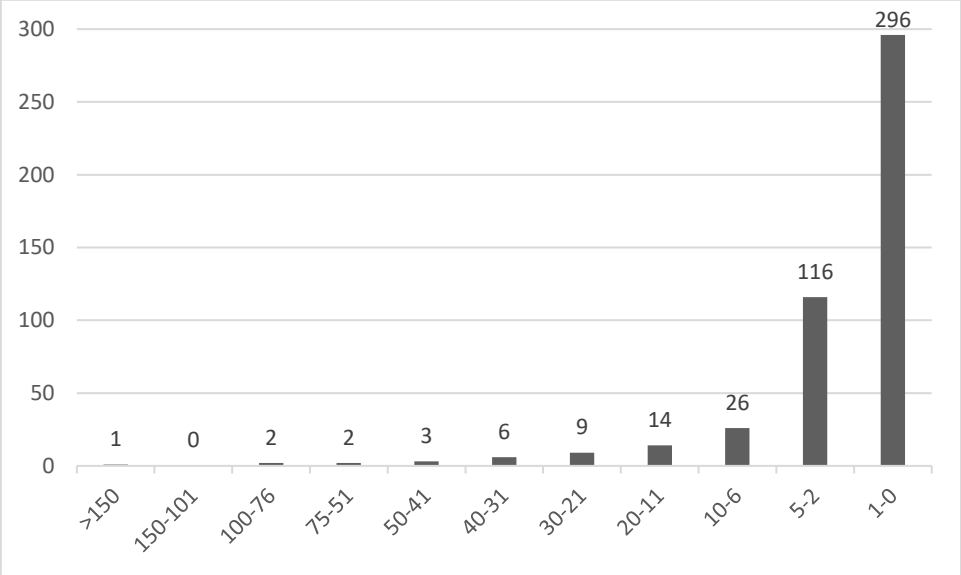

Figure 9 – Adblock Crawl number of websites distribution

Regarding the Ghostery crawl, there were 2.075 cookies belonging to 271 different entities, it is necessary to identify the third-parties, in Table 14 will be shown the top 10 third-parties of this execution and in the following image their distribution by website. Is necessary to mention 'sapo.pt' is present in the Top 10 third-party trackers.

Table 14 – Ghostery Crawl top 10

Type	Number	Percentage
google.com	87	12,43%
cloudflare.com	40	5,71%
youtube.com	30	4,29%
accounts.google.com	23	3,29%
sapo.pt	13	1,86%
onesignal.com	12	1,71%
webcare.byside.com	10	1,43%
superinterstitial.com	6	0,86%
imgur.com	6	0,86%
yahoo.com	5	0,71%

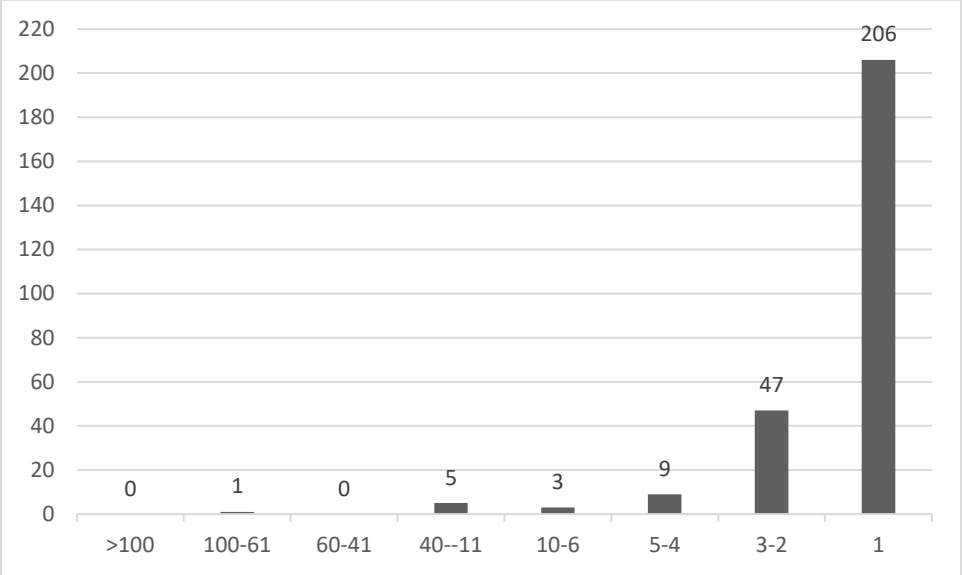

Figure 10 – Ghostery Crawl Number of websites distribution

In conclusion, the hosts or entities behind the third-party cookies tend to belong to the same companies, despite the reduction of the number of cookies with both extensions.

Ghostery, prevented doubleclick.net to create any cookies on its crawl, even tough, it had reached 33,71% of the websites on the first crawl and 24,00% in the ABP crawl.

The latter extension was so efficient in removing popular trackers that a Portuguese tracker reached the Top 5 of the global trackers.

5.3 Prominence

The rank in the section above focus only on the scale: if third-party is present in more websites it should be higher on the rank. However, this section will show a different approach to calculate third-party rankings.

Instead of just comparing the number of websites the tracker is present it will evaluate the sites in which they are present, for example, a third-party that appears only on one site but is the 5th most visited website, in contrast to, a third-party which appears only in the least visited website, it will have different rankings.

In order to achieve this, it was used the prominence metric proposed by the authors of OpenWPM, $Prominence(t) = \sum edge(s,t) = 1\frac{1}{rank(s)}$, where t is a third-party and s the site being visited.

The formula above measures the frequency with which the online user will be exposed to a given third-party. It is also especially useful when identifying the most important third-parties

because it provides a more reliable comparison in contrast to just comparing a simple prevalence count.

Additionally, it can be used to both analyze the efficiency of tracking protection tools and the growth of the web trackers over time, and correlate their presence cross markets.

Taking into consideration the prominence formula, Table 15, compares prominence versus prevalence, of the top 10 third-parties for the crawl without any privacy configuration. The table below also shows the difference in prominence rank in contrast to simple prevalence.

Table 15 – Crawl prominence vs prevalence

Third-party	Number of Sites	Prominence (Pro)	Prevalence (Pre)	Difference (Pro-Pre)
doubleclick.net	236	1,74	1	=
adnxs.com	113	1,29	5	+3
google.com	206	1,11	2	-1
google.de	120	0,87	4	=
scorecardresearch.com	108	0,86	6	+1
rubiconproject.com	91	0,84	7	+1
yahoo.com	57	0,70	13	+6
advertising.com	34	0,68	32	+26
hit.gemius.pl	32	0,67	39	+30
rlcdn.com	51	0,65	19	+9

To analyze the efficiency of tracker protectors, the Figure 11 shows a plot of the prominence compared in the three crawls, demonstrating that the tracking the user is subject is much smaller with extensions, specially Ghostery.

It's significant that the results obtained shows these tools are more efficient against trackers present in popular sites, in contrast to, trackers present in obscure sites, this information is supported by Table 14 where it's found on the top 10 third-parties a Portuguese third-parties.

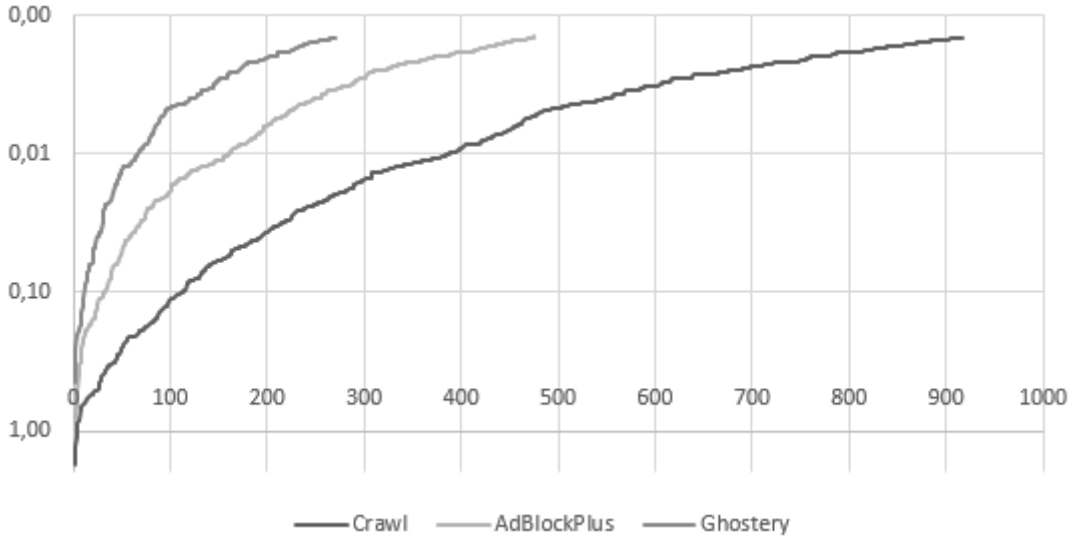

Figure 11 – Prominence Comparison

5.4 Third Parties per Website

Another useful information is the reverse of previous topics, instead of identifying the most important third-parties it will identify the websites which contain the most third-parties. This allows the distinction of the global ranking and the subset of the sites which have as a top-level domain '.pt'.

Of the initial list 700 websites, there are 196 which end in '.pt' since this study focus on the Portuguese market making this validation is necessary, however, is worth mentioning there are several Portuguese sites that don't have the country suffix.

For the initial crawl, Table 16 and Table 17, show the websites with the most third-party cookies, one showing the global crawl and the other focusing only on the websites ending in '.pt'.

Figure 12 illustrates the distribution of the number of third-parties by website, while Figure 13 focus only in the '.pt' websites. Is significant that in both cases there are several websites which don't contain any kind of third-party. For the global crawl, 39% of the visited sites don't have any third-party and in the '.pt' subset the number is 45.4%

Table 16 – Crawl Top 10 websites with more third-parties

Website	Number
lifebuzz.com	84
futhead.com	80
dailymotion.com	79
speedtest.net	64
nba.com	61
clipconverter.cc	60
photobucket.com	55
dailymail.co.uk	55
flytap.com	54
cnet.com	53

Table 17 – Crawl Top 10 website '.pt' with more third-parties

Website	Number
sapo.pt	50
homeaway.pt	47
tempo.pt	45
edreams.pt	40
iol.pt	37
jn.pt	36
kuantokusta.pt	36
sabiasque.pt	33
dioguinho.pt	28
sportzone.pt	25

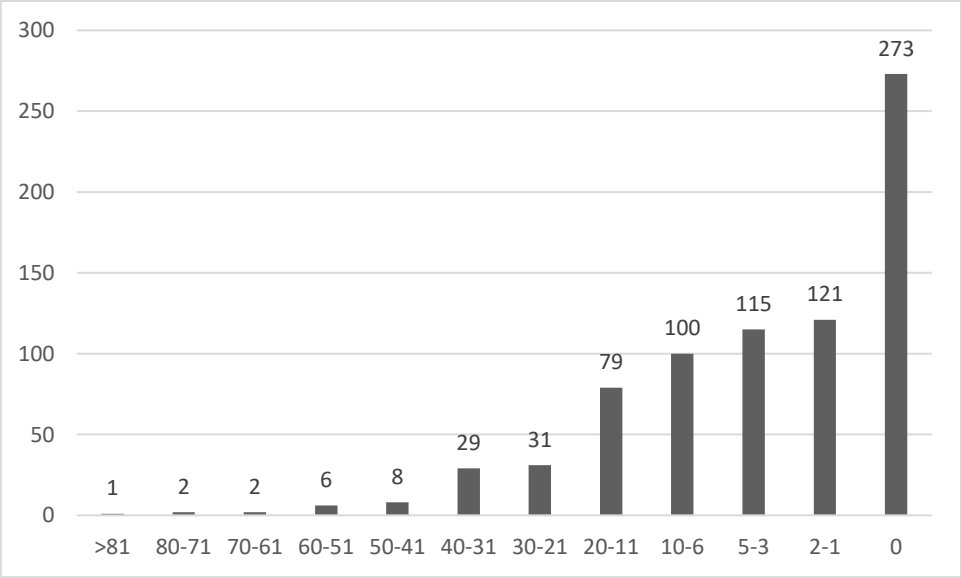

Figure 12 – Crawl Number of third-party distribution

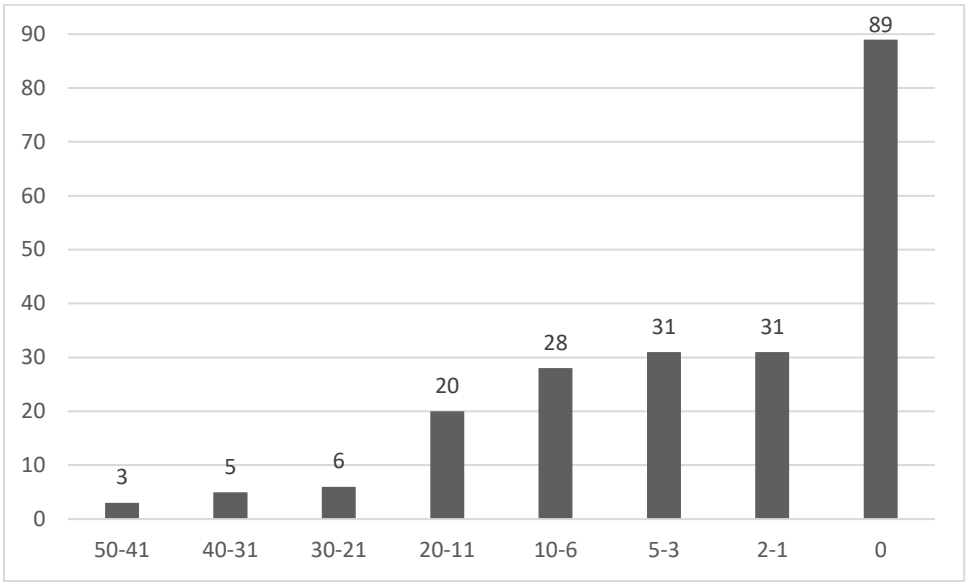

Figure 13 – Crawl Number of third-party distribution in '.pt' websites

For the ADP crawl, Table 18 shows the websites with more third-parties of the full data set. On the other hand, Table 19, will provide the same information but only for websites which end in '.pt', like what was done for the first crawl.

Table 18 – Adblock Crawl Top 10 websites with more third-parties

Website	Number
microsoftstore.com	24
sharepoint.com	22
tempo.pt	18
businessinsider.com	18
tripadvisor.com.br	16
baixaki.com.br	16
nba.com	16
gamespot.com	16
bolsademulher.com	16
dropbox.com	16

Table 19 – Adblock Crawl Top 10 website '.pt' with more third-parties

Website	Number
tempo.pt	18
kuantokusta.pt	15
edreams.pt	14
dioguinho.pt	13
jornaldenegocios.pt	13
hugogil.pt	11
conviviocm.pt	10
meo.pt	10
publico.pt	9
tabonito.pt	8

The Figure 14 explains the distribution of third-parties by all websites and as mentioned before, in the data sample there are 196 websites with '.pt' as top level domain, Figure 15, exposes the dissemination of the third-parties in those websites.

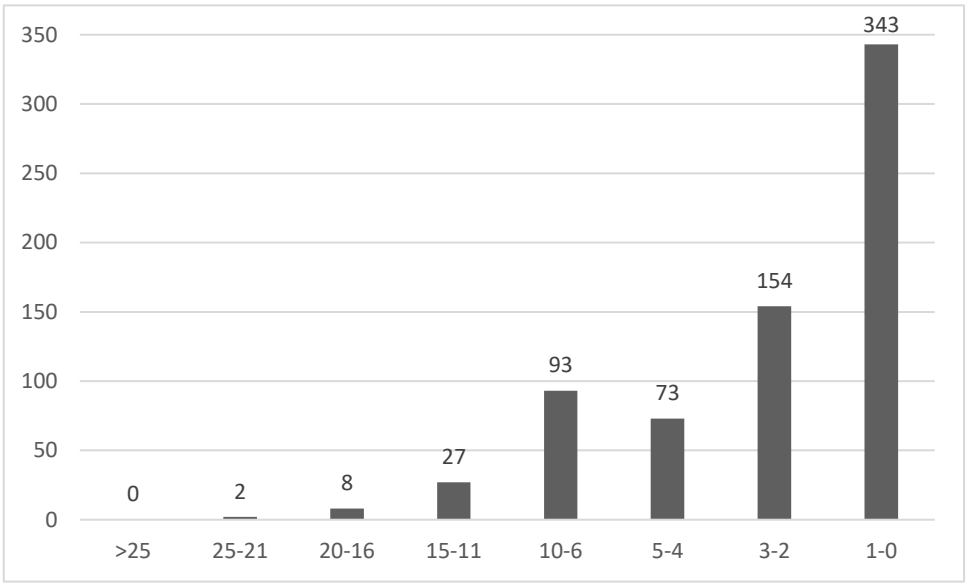

Figure 14 – Adblock Crawl Number of third-party distribution

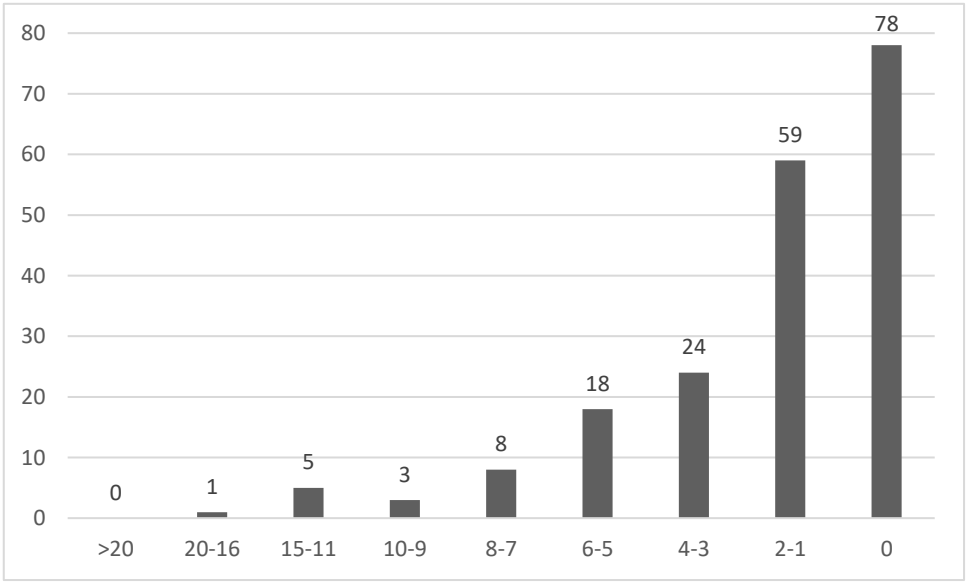

Figure 15 – Adblock Crawl Number of third-party distribution in '.pt' websites

For the last crawl, the one with Ghostery extension, it will be represented the 10 most websites with more trackers on Table 20. On Table 21 it will be shown the 5 most, limited to the sites ending in '.pt'.

Table 20 – Ghostery Top 10 websites with more third-parties

Website	Number
seriesparaassistironline.org	10
seriesonlinehd.org	8
hugogil.pt	8
tafeio.tv	7
delas.pt	6
mobafire.com	6
mercadolivre.com.br	6
sharepoint.com	6
imzog.com	5
hdzog.com	5

Table 21 – Ghostery Crawl Top 5 website .pt with more third-parties

Website	Number
hugogil.pt	8
delas.pt	6
publico.pt	5
dinheirovivo.pt	5
zerozero.pt	5

The next image will demonstrate the distribution of third-parties by website, while the following will focus only in websites ending in '.pt'. With this extension, there are 40,57% websites which use third-party cookies and on the Portuguese top level domain the number is 34,18%.

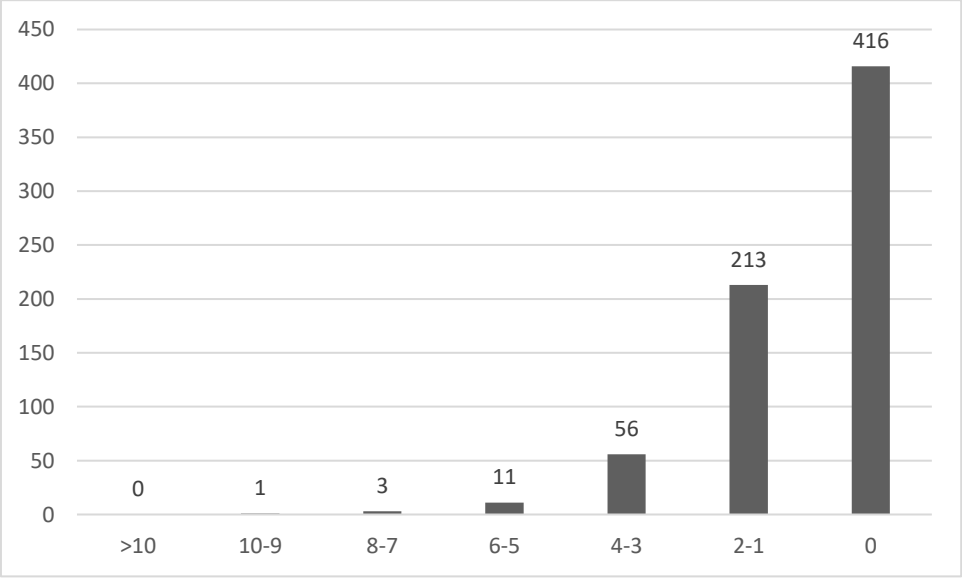

Figure 16 – Ghostery Crawl Number of third-party distribution

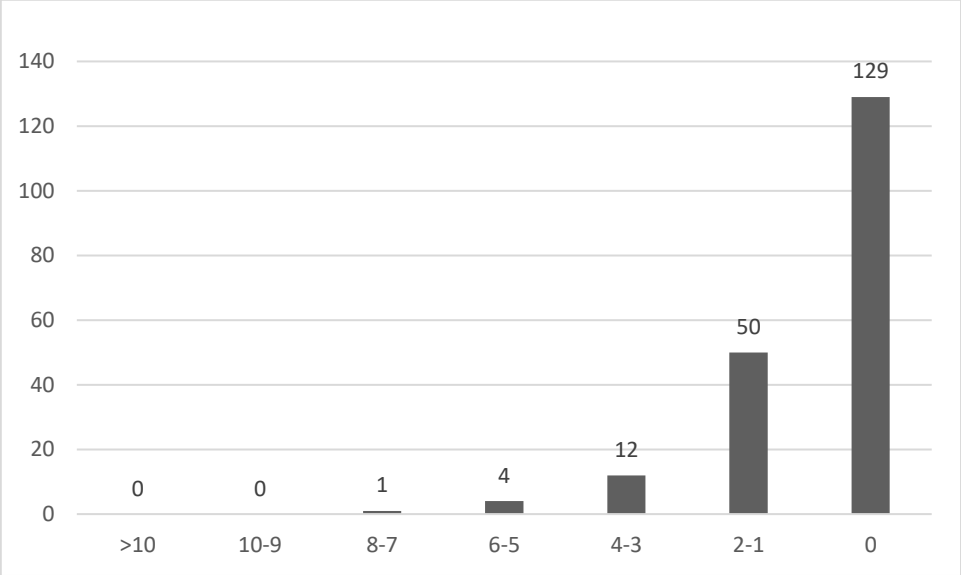

Figure 17 – Ghostery Crawl Number of third-party distribution in '.pt' websites

5.5 Tracking

In this section, it will be detailed how the third-party tracking takes place and the entities behind it. The identification of tracking will use the methods described in 4.3.4 to obtain the cookies which share the same host, key and value cross websites.

Starting with the crawl without any additional privacy configuration, it was gathered all the cookies which share the characteristics above and the result was 1.639 cookies.

This results, detailed in Table 12, shows that from the top 10 tracking capable cookies 7 belong to Google companies reaching as much as 236 different websites. This represents slightly more than a third of the visited websites.

The identifying cookies most of the times doesn't even try hide the fact it is directly or indirectly identifying the user with the cookies key using the value "ID" or variants. The value of the cookie is a long alphanumeric string without any apparent meaning. For example, for the "doubleclick.net" third-party present in 236 different websites, the key is "ID" and the value is "222bfc1e8109002f||t=1473616764|et=730|cs=002213fd4839d229357359dce3".

Focusing only on websites which end in '.pt', Table 17, there are 473 which contain tracking capable cookie. From the Top 10, Google companies take the top 6 being able to reach up to 57 different websites.

Moving to the ADP crawl the number of tracking capable cookies is 455. Analyzing this Top 10, Table 13, Google companies take up 4 spots reaching up to 168 unique websites, the addition of twitter which now takes 3 spots in top 10 stretching up to 70 websites.

In the Portuguese top level domain, there are 104 tracking capable cookie. The Top 10, Table 19, of this market is dominated by Google companies taking all the Top 5 spots. Additionally, in that ranking there is the first Portuguese third-party, sapo.pt, stretching to 12 websites.

Lastly for Ghostery crawl, using the same approach as before, the first step to do is to identify the cookies which can contribute to online tracking and the result is 86. From these cookies, the Top 10 more predominant, Table 20, Google companies still take 4 spots, and Sapo.pt gets the 6th position of the global tracking reaching 13 websites.

Focusing only on the top-level domain '.pt', Table 21, the number of cookies is 18, which represents a reduction of 96,19% comparing with the initial analysis. On the top 3, it was found 2 Google companies reaching 22 websites and sapo.pt reaching 12 websites.

In the initial run, it was obtained 1.639 tracking capable cookies then on the ADP crawl it was obtained 455 which accounts for a reduction of 72,24% and on the Ghostery crawl where there were only 86 tracking capable cookies, which is a reduction of 94,75%.

6 Conclusion

For this dissertation, the Portuguese web is analyzed to identify which are the main third-parties, entities which the user is not aware while browsing a website, and which methods and technologies those entities use to track and gather information.

It was described with detail a list of known tracking methods and a list of tools available which can protect the user from less sophisticated tracking mechanisms. It was also detailed frameworks which are able to measure privacy online.

Based on a list of the 700 most visited websites by Portuguese users, of those 196 had the top-level domain '.pt', although due to the liberalization of top level domains is possible to have Portuguese sites in other top level domains.

The websites were distributed in 38 categories; File sharing was the most common category with 61 websites followed by news with 56.

Three main crawls were executed, one without any privacy setting, one with the extension Adblock Plus and one last with the extension Ghostery. The results obtained showed without any protection it created up to 48.876 cookies, the ADP extension reduced the number by 59,35% and Ghostery further decreased to 84,02%.

Related with the reduction of cookies, the extensions also had a significant impact in the JavaScript cookies reducing it by 44,87% and 84,02% respectively. In regards to profile cookies, was the type of cookies which had the biggest reduction with Adblock Plus reducing the creation of this type of cookies by 74,30% and Ghostery reduced the number in a stunning 91,13%.

Regarding the third-parties, on the first execution upon visiting the 700 websites it was created 29.081 third-party cookies. This number was reduced to 7.230 with the extension Adblock Plus, 75,14% reduction. On Ghostery crawl the number of third-party cookies identified was 2.075 this is equivalent to a reduction of 92,87% of third-party cookies created.

To identify the third-party which have the biggest overview of the user online habits, it was used the prominence metric, which takes into consideration the rank of the site which contains the third party, it also compared the prevalence with the number of sites on which the entity was present. The comparison of the results of these two methods allowed to identify the major third-party in Portuguese web, Google but also another less known, adnxs.

Focusing on the websites, it was identified the websites which contained the most third-parties, and since this study focus on the Portuguese market, the sites with the Portuguese top-level domain had a separate analysis. However, with all three crawls and the separation of the full list of sites and only the '.pt' sites it wasn't identified a specific category which had more third-parties than the others.

In regards to tracking, the unblocked crawled obtained 1.639 tracking capable cookies then on the Adblock Plus crawl it was obtained 455 which accounts for a reduction of 72,24% and on the Ghostery crawl there were only 86 tracking capable cookies, which is a reduction of 94,75%.

It is clear the web is full of third-party cookies and users without any sort of protection are very vulnerable to being tracked, fortunately, as it was demonstrated the tools freely available on the web to reduce tracking do have a very significant impact in reducing tracking and overall the presence of all types of cookies be it first or third-party, JavaScript or profile cookies.

6.1 Recommendations

Unfortunately, total online tracking prevention doesn't seem possible, but there are steps the user can and should take in order to regain some of it. First of all, as demonstrated in the results of this analysis the freely available tools do have a very impactful result on the exposure to cookies.

As was learned from the state of art, the user should actively clean cookies using appropriate tools in order to clean all sorts of data storage vectors available to the website.

6.2 Contribution

In this section, it will be presented the goals defined in the beginning of the document, 1.3, and if it was possible provide an answer to them based on the results of the analysis.

- Which are the most predominant third-parties?

In this research, it is provided two ways of identifying the most predominant third-parties: prevalence count and prominence. The first simply looks at how many sites the third-party is present and ranks it, the second uses a slightly more elaborate formula by considering the position the website has in top 700 websites used in this study.

It is worth mentioning the presence of three Google domains in the top 10 of third-parties.

Table 15, illustrates the comparison of both methods of identifying the most predominant third-party and comparing prevalence with prominence.

- What the third-parties do with the information gathered?

Finding an answer to what companies do with the user information was extremely difficult to correctly obtain, despite the existence of tracker repositories lists. However, there isn't information available to evaluate what is being done with the user information.

Based on visiting the third-party web pages, what was possible to gather is that most, of the sample visited, sell the information to websites in order to identify the main areas of interest of the users.

Nevertheless, this was goal wasn't achieved.

- In which websites are third-parties present on?

The base of this research was the top 700 websites of the Portuguese market, which include sites from all around the globe, regardless, it was of interest to look specifically to sites with the top domain '.pt'.

Based on the cookies associated to a website, it was determined the unique hosts, different from the website which they belong, and that information was detailed in Table 16 and Table 17. No pattern was found regarding the number of cookies created and the website category.

- How does the cookie tracking work?

Cookies have the following structure: Domain, Key, Value and Host. Tracking occurs when the same key, value and host appear in several websites. With this the third-parties can follow users throughout websites and session.

The field value of a cookie is generally a long alphanumeric string without any apparent meaning. Initially, it was created 1.639 cookies which share the same key, value and host with at least one more cookie. In the ADP crawl the number reduced to 455 cookies and Ghostery reduced the number to 86, this is a reduction of 94,75% compared to the initial run.

- How can third-parties be avoided?

During the research, it was clear that it wasn't possible to completely avoid being tracked by third-parties, so the next objective was to identify how to reduce it. To achieve this two extensions were tested: Adblock Plus and Ghostery.

It was executed 3 crawls, one without any anti tracking tools, one with ABP extension and one with Ghostery. The numbers of cookies created with these tools were reduced by 59,35% with ADP and 84,02% by Ghostery.

Even though the goal wasn't totally achieved, because is not possible, the results obtained with ADP and Ghostery are, still, good indicative that the user may protect himself from trackers.

6.3 Future Work

It was identified for future work, correlate the JavaScript executed in the web page being visited with the information being stored on the cookies, with the goal of identifying how the web trackers can identify users and identify the methods a specific tracker uses.

Secondly, validate the impact of the setting 'Do Not Track' has on the number of third-party present and on the number of identical cookies cross website. This setting is one of the few options the user can set to try to reduce online tracking without installing additional extensions and software.

Lastly, adapt the analysis method developed in Java to the OpenWPM framework in python, to help other researchers analyze the results obtained and easily compare them between markets.

References

Acar, G. *et al.*, 2014. The Web Never Forgets. In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security - CCS '14*. New York, New York, USA: ACM Press, pp. 674–689. Available at: https://securehomes.esat.kuleuven.be/~gacar/persistent/index.html.

ANACOM, 2015. ANACOM - Serviço de acesso à Internet. Available at: http://www.anacom.pt/render.jsp?contentId=1372507#.VrtsZvmLTDd [Accessed February 10, 2016].

Anon, 2014. ARTICLE 29 DATA PROTECTION WORKING PARTY. Available at: http://ec.europa.eu/justice/data-protection/index_en.htm [Accessed October 19, 2016].

Ayenson, M. *et al.*, 2011. Flash Cookies and Privacy II: Now with HTML5 and ETag Respawning. *SSRN Electronic Journal*. Available at: http://www.ssrn.com/abstract=1898390.

Bernal, P., 2013. Google, privacy and a new kind of lawsuit. *2013-01-28*. Available at: https://paulbernal.wordpress.com/2013/01/28/google-privacy-and-a-new-kind-of-lawsuit/ [Accessed September 19, 2016].

Castelluccia, C., Castelluccia, C. & Castelluccia, C., 2013. Data Harvesting 2 . 0 : from the Visible to the Invisible Web. Available at: https://hal.inria.fr/hal-00832784.

Datta, A., Tschantz, M.C. & Datta, A., 2015. Automated Experiments on Ad Privacy Settings. *Proceedings on Privacy Enhancing Technologies*, 2015(1), pp.92–112. Available at: http://www.degruyter.com/view/j/popets.2015.1.issue-1/popets-2015-0007/popets-2015-0007.xml.

Eckersley, P., 2010. How unique is your web browser? *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 6205 LNCS, pp.1–18. Available at: https://panopticlick.eff.org/static/browser-uniqueness.pdf.

Englehardt, S. *et al.*, 2015. OpenWPM : An automated platform for web privacy measurement. Available at: http://senglehardt.com/papers/openwpm_03-2015.pdf.

Englehardt, S. *et al.*, 2014. Web privacy measurement: Scientific principles, engineering platform, and new results. *Manuscript posted at http://randomwalker. info/publications/WebPrivacyMeasurement. pdf*. Available at: http://randomwalker.info/publications/WebPrivacyMeasurement.pdf.

European Parliament, 2012. EUR-Lex - 32002L0058 - EN. *Official Journal L 201 , 31/07/2002 P. 0037 - 0047;*

Fiegerman, S., 2013. Disconnect.me Lets You Control Your Data Online. Available at: http://mashable.com/2013/04/17/disconnect-me/#GGw7IV0vfgq5 [Accessed January 31, 2016].

Ghosh, A. & Roth, A., 2013. Selling privacy at auction. *Games and Economic Behavior*, 1, pp.1–13. Available at: http://dx.doi.org/10.1016/j.geb.2013.06.013.

Heckel, P.C., 2013. How To: Use mitmproxy to read and modify HTTPS traffic - Philipp's Tech Blog. Available at: https://blog.heckel.xyz/2013/07/01/how-to-use-mitmproxy-to-read-and-modify-https-traffic-of-your-phone/ [Accessed January 31, 2016].

Liu, B. *et al.*, 2013. AdReveal. In *Proceedings of the Twelfth ACM Workshop on Hot Topics in Networks - HotNets-XII*. New York, New York, USA: ACM Press, pp. 1–7. Available at: http://www.technicolorbayarea.com/papers/2013/LSWCG13adreveal.pdf [Accessed February 14, 2016].

Mayer, J.R. & Mitchell, J.C., 2012. Third-party web tracking: Policy and technology. *Proceedings - IEEE Symposium on Security and Privacy*, pp.413–427.

Mowery, K. *et al.*, 2011. Fingerprinting Information in JavaScript Implementations. *Web 2.0 Security & Privacy*, pp.1–11. Available at: http://cseweb.ucsd.edu/~hovav/papers/mbys11.html.

Mowery, K. & Shacham, H., 2012. Pixel Perfect : Fingerprinting Canvas in HTML5. *Web 2.0 Security & Privacy 20 (W2SP)*, pp.1–12. Available at: https://cseweb.ucsd.edu/~hovav/dist/canvas.pdf.

Oracle, 2015. The Set Interface. Available at: https://docs.oracle.com/javase/tutorial/collections/interfaces/set.html [Accessed September 15, 2016].

Python Software Foundation, 2016. sqlite3 - DB-API Interface for SQLite databases. Available at: https://docs.python.org/2/library/sqlite3.html [Accessed September 15, 2016].

Roesner, F., Kohno, T. & Wetherall, D., 2012. Detecting and defending against third-party tracking on the web. *Proc. of the USENIX Conference on Networked Systems Design and Implementation (NSDI)*, (Nsdi), p.12.