Bachelor thesis
Computer Science

Radboud University

Web Tracking And Current Countermeasures

Author:
Willem Boumans
S4337166

Supervisor/assessor:
Dr. Ir. Erik Poll
e.poll@cs.ru.nl

April 5, 2017

Abstract

Tracking users on the web is widespread nowadays. All kinds of techniques are employed to follow users on the internet. Not just cookies, but all sorts of other tracking and fingerprinting methods are used to do so. Many users delete their cookies frequently, use their browser's private mode or use certain browser plugins in order to reduce the extent to which they are tracked.

Being tracked however can never be completely avoided. A number of tracking methods exist, that are very hard or nearly impossible to block. Most of these methods are results of the cat and mouse game between tracking parties and privacy-conscious users.

In this thesis, we will look into the way web browsers and privacy extensions counter known tracking methods and the shortcomings they have. We found out that web browsers all apply similar techniques to block web trackers. The same is true for privacy enhancing browser extensions, which almost all rely on blacklists to block trackers. Currently, only Privacy Badger uses a different approach, by using algorithms to identify trackers. In my eyes, this is a promising feature that could be better than blacklists.

Contents

Chapter 1

Introduction

The internet has come a long way since its "static" period, in which user interaction was non-existent. More and more new concepts were added in order to make the internet more dynamic. This of course added a lot of complexity and potential security or privacy risks as well.

These additions gave web site owners quite a lot of new features for their web pages, including the tracking of users. One of the key techniques for this are HTTP cookies, although not all tracking mechanisms make use of them. Many alternatives are prevalent because HTTP cookies can easily be disabled or deleted by the user. Alternatives for HTTP cookies mostly consist of storing an identifier in a harder to reach and, most importantly, harder to delete storage location. Another common tracking method is the fingerprinting of users and their devices. Here an identifier is made using data that a user "leaks" while browsing, such as their IP address, installed fonts in a browser or their typing habit. These techniques will be further explained in Chapter 2.

Tracking purely with HTTP cookies would be the most transparent way to track users, but trackers found this method to be not persistent enough. Commercial interests of tracking parties makes them want to track users more precisely and more persistenly [1]. This is getting easier and harder at the same time: easier since more and more tracking techniques are invented, harder because users use more and more different devices to browse the web and because they apply private browsing and privacy enhancing plugins more and more [33]. In the game of cat and mouse that is web tracking, trackers always try to be a step ahead of consumers.

This raises the question: what options are available to reduce the level of being tracked and which tracking methods cannot (yet) be avoided? With all popular browsers offering privacy settings and a range of privacy-enhancing tools being available, one would think that trackers on the web could be almost completely disabled.

Of course, there might be solutions to counter every known tracking method, but they are of no use if they are unfindable, hard to understand or limit usability severely. Therefore, we will limit our research to countermeasures against web tracking that are easily available and have a high user base. This means that options that are offered "out of the box" by web browsers are very interesting to look into, as are privacy enhancing plugins since they are easilly installable and do not need to be configured much.

We would like to find out what the current balance is between online tracking and privacy protection options. To gain an insight in this balance, we will make an overview of the tracking techniques that are in use, what level of tracking they offer and of course of the ways these techniques might be countered or hindered by privacy-increasing methods. As mentioned earlier, we will mainly focus on techniques (both tracking and tracking protection) that are used by many users.

The expectation is that most of the "older" and more known tracking methods can be effectively countered by both browser privacy settings and external add-ons. More obscure trackers or fingerprinting techniques will most likely be a lot harder to counter, while some methods might not be counterable at all as of now. This means that web browsers will most likely not have options in them to block these tracking methods.

The base for this hypothesis is former research conducted in this domain. A key research on tracking and fingerprinting on the web is the `panopticlick`[1] research [18] in which the EFF (Electronic Frontier Foundation) showed that over 90% of web browsers can be uniquely fingerprinted by looking at installed fonts in the browser. Hoofnagle *et al.* concluded that there is a large growth in the usage of tracking cookies and that new, unavoidable tracking mechanisms have been developed [21]. Mayer and Mitchell present issues on policy and technology with third party tracking [28]. Roesner *et al.* created a client-side method for detecting third party trackers. They found out that web trackers can capture a significant amount of a user's behaviour. Bujlow *et al.* conducted a similar research to this thesis [11], by presenting an overview of used tracking methods and countermeasures against them. The difference however, is that Bujlow *et al.* present countermeasures against the tracking options they mention, instead of covering the most popular and present options. The timeline of web tracking they present, gives a simplified overview of the evolution of tracking [11, Figure 1].

Fellow Radboud students Ivar Derksen and Patrick Verleg based their bachelor theses [17, 44] on HTML5 tracking and tracking with cache storages in web browsers respectively. Derksen concluded that browsers do not take expiry dates into account with IndexedDB and Local Storage. Private

[1] `https://panopticlick.eff.org`

browsing deleted Local Storage entries after browsing, but IndexedDB is not treated as such for all browsers. Verleg found out that cache cookies (identifiers stored in some form of browser cache) are very common and that it is hard for an average user to prevent being tracked on the web.

In Chapter 2, an overview of tracking techniques that are currently being used is given. Here, we distinguish between tracking methods using some form of data storage and tracking users and devices by fingerprinting. For methods using data storage, we will look into the most popular methods. Fingerprinting techniques are split up into active and passive fingerprinting.

Chapter 3 presents countermeasures against web tracking that are implemented in web browsers. These include the settings offered in web browsers and private browsing mode, but also include regulation and the Do Not Track and P3P incentives.

During the research, it became clear that it is necessary to have a clear understanding of the difference between a local observer (or attacker) and a remote one. This is needed because some of the privacy measures we discuss focus more on the local observer instead of a web tracker, meaning their goal is different than that of privacy measures focussing on remote trackers. The difference in approaches against either a local or remote observer is covered in section 3.3.1

Chapter 4 is about privacy enhancing browser extensions. Both privacy tools such as Ghostery and Privacy Badger and ad blockers are covered in this chapter. After presenting the countermeasures gainst web tracking that are offered in browsers or in extensions, we will compare the implementation of them in browsers and/or extensions. From this, we can determine in what way privacy settings and extensions really enhance the privacy of users.

Chapter 2

Current tracking methods

This chapter lists a range of tracking mechanics that are currently being used on the web and explains how they work. Techniques using data storage will be covered in section 2.1. Fingerprinting will be covered in sections 2.2.1 (passive fingerprinting) and 2.2.2 (active fingerprinting). Countermeasures against tracking techniques will be presented in chapter 3 and 4.

2.1 Tracking techniques using data storage

The most common way to track users across the internet is by storing unique identifiers somewhere on their computer. These can then be queried by tracking parties, following users around the web with their identifier. The difference with fingerprinting (covered in section 2.2) is that fingerprinting does not store an identifier locally on the user's computer and that fingerprinting relies on the attributes of a user, such as his IP address or the way his GPU renders a picture. Tracking using data storage works with identifiers that are created by a tracking party and do not have anything to do with the user's specific attributes.

2.1.1 HTTP cookies

The most known way to track users is by utilizing HTTP cookies, often referred to as just "cookies". Web browsers rely on the HTTP (Hyper Text Transfer Protocol) protocol to transfer information [37]. HTTP, however is a stateless protocol, which means that no session information is kept by either the server or the client. This made browsing with services for which a browser state needed to be kept (such as the logged in user) impossible. As a solution, it was made possible to save small text files (up to 4KB) on a users computer, in which the browser state was stored [7]. These text files are sent with each HTTP request message. This concept later became known as cookies.

HTTP cookies are set using the `Set-Cookie` header. This header is included in an HTTP response from a server. A `Set-Cookie` header makes the browser store a cookie on the users computer and this cookie will be sent with any request to the server that issued the cookie.

Cookies with an expiration date are saved on the users computer until the expiration date passes or until the cookie is deleted. When no expiration date is specified while setting the cookie, it is regarded as a session cookie. This means the cookie will be deleted once the browser session is terminated. Other keywords for cookies are `Domain`, `Secure` and `HttpOnly`. These specify the domain to which the cookie belongs (and thus to which servers the cookie can be sent back by the client), whether the cookie may only be sent over a secure connection and whether or not the cookie may be accesed by scripts running client side.

Apart from the HTTP `Set-Cookie` header, cookies can also be set (and read) by JavaScript, by interacting with the Document Object Model (DOM)[1].

HTTP cookies can either be "first party" or "third party". First party cookies are cookies set by the domain a user is *directly* visiting, while third party cookies are set by domains that are indirectly visited, for instance when the first party loads a resource from the third party. An example would be `firstparty.com` which has included an image from `thirdparty.com`. When a user visits `firstparty.com`, cookies set by `firstparty.com` are first party cookies, while cookies from `thirdparty.com` are third party cookies.

Usage in tracking

HTTP cookies can be used as a tracking tool in numerous ways. They can be used on their own, by just placing a cookie with an identifier on the user's device when he/she visits the web page, or they can be used in combination with other techniques. These include cookie syncing, cookie respawning and tracking aggregators [11]. Cookie syncing means cookies from one domain are passed on to another one. Respawning cookies brings back deleted cookies from another storage location, in which the same identifier was stored. This will be further elaborated in section 2.1.2. Some tracking parties serve as aggregator for other tracking services. The tracking party then sends request to the aggregator, which include the identifier stored in the cookie placed by the tracking party. This means that the aggregator will collect the identifiers set by many separate tracking parties.

Not all HTTP cookies are used for tracking purposes, but Li *et al.* have showed that tracking and non-tracking cookies can be distinguished with a very high accuracy [26].

[1] `https://www.w3.org/DOM/`

2.1.2 Flash cookies

Adobe Flash uses Local Storage Objects (LSOs) to store data in. These so-called Flash cookies are used by Flash applications to store local data used by them. Flash cookies are also used to track web users. They offer some advantages over HTTP cookies for tracking parties. By default, Flash cookies offer a storage of 100 KB. This can be extended to an "infinite" amount, when allowed by the user. This is thus at least 25 times larger than the 4 KB of storage HTTP cookies offer. The extra storage might be useful for storing more information about users, but unique identifiers are easily storable in 4KB as well. Most importantly, Flash cookies are stored in a different and more hidden away location than HTTP cookies. Only since Flash Player 10.3, Adobe offers a way to clear Flash LSOs, with the *ClearSiteData* API. Most current browsers make a call to this API when their "clear-cookies" functionality is used [11]. This will covered in section 3.2.

Another difference between Flash cookies and their HTTP counterparts, is that Flash cookies do not expire by default. In fact, they do not have an expiration-property at all. This means that the Flash LSOs stay on the user's computer until they are deleted. Finally, Adobe Flash does not have a separate storage for each browser the plugin is installed on. This means that Flash cookies set in a browsing session with one browser, can be accessed in another browser and thus that users can be tracked across browsing platforms with flash cookies.

Usage in tracking

Flash cookies are used to track users in a similar way to HTTP cookies. That is, an identifier can be stored as a LSO. This identifier can later be accesed by web pages containing Adobe Flash elements. These elements need not be visual elements, so this Flash application can possibly be running on the background, hidden out of sight from the user.

Cookie respawning

In 2010, Soltani *et al.* indicated that Flash cookies were used to "respawn" HTTP cookies [42] . Identifiers which were initially stored in HTTP cookies, were also stored in Flash cookies. Since the Flash cookies were not deleted when HTTP cookies were removed from the system, the identifiers stayed stored on the computer. This made it possible to set a new HTTP cookie, with the original identifier by getting this from the Flash storage and thus tracking users across multiple sessions. In 2011, Ayenson *et al.* show that HTTP cookies were not only respawned with Flash cookies, but also using HTML5 storage functionality and ETags [5].

The same year, McDonald and Cranor surveyed 500 websites to check if the respawning still occured [29]. No respawning was encountered in 500 randomly selected sites, but from 100 most visited sites, at least two web pages were respawning HTTP cookies with Flash cookies. In 2014, Acar *et al.* created an automated test to check if websites respawned cookies using Adobe Flash [2]. This research checked 10.000 websites and found that 33 Flash cookies from 30 parties respawned 355 HTTP cookies on 107 domains which acted as first parties.

Respawning cookies is not something limited to Flash LSO, HTML5 or ETags. In principle, every storage location which is accessible by a web page (most likely with scripts running on them), can and might be used to store identifiers, later to be used in cookie respawning.

2.1.3 Silverlight Isolated storage

Microsoft Silverlight also offers a storage which their applications can use. This "Isolated storage" offers each web page with Silverlight elements 100 KB of storage space, which do not necessarily have to be key/value pairs. Silverlight is, contrary to most of the other plugins mentioned, going to disappear quite soon. In 2021, support from Microsoft for Silverlight will stop[2]. In 2011 the development already stopped in favour of HTML5 techniques.

Silverlight Isolated storage offers some ways to remove entries in their storage. This can be done by deleting files from a hidden folder, or you can use the Silverlight options panel. This panel is only accesible if Silverlight is running visually on an opened website, by right-clicking on the UI of the application. A browser API such as for Flash LSOs does not exist. Via this panel from Silverlight, the storage can also be disabled entirely. Since an API for cleaning Silverlight storage is lacking and since Silverlight storage can only be cleaned with a relatively hard to acces panel, Silverlight storage has the possibility to be more persistant than Flash cookies. This makes Silverlight's Isolated storage more "dangerous" for privacy in a way.

Usage in tracking

Silverlight storage can in theory be used the same way as Flash or HTML5 storage. This means all the threats coming with a plugin offering storage, hold. These include the possibility for cookie respawning. A difference between Silverlight and Flash or HTML5 is that Silverlight has never been supported for Linux systems by Microsoft. The expectation is that tracking via Silverlight is becoming less and less prevalent, since the application itself is too.

[2] `https://blogs.windows.com/msedgedev/2015/07/02/moving-to-html5-premium-media`

2.1.4 HTML5 Local storage and IndexedDB

HTML5, introduced in 2014, tries to improve the support for web applications with user interaction. This new standard also came with some new storage locations and thus locations to store tracking identifiers. The two main storage options HTML5 offers are Local storage and IndexedDB.

Local storage in HTML5 is part of the *WebStorage* API. Storage objects are stored as a key/value pair, which is similar to HTTP cookies. The objects are however larger (at least 5 MB) and information is not automatically sent to a server, but a web page must request it itself. Local storage also has a more temporal form: Session storage. Session storage for a certain session is removed once a browser tab is closed. This means that multiple tabs showing the same web site, will not be able to acces each other's Session storage. This is where HTML5 Session storage really differs from HTTP cookies, since non-persistent cookies will be kept until the entire web browser is closed. Since Session storage is only kept very shortly, there is no real use in tracking users with this storage location.

HTML5 Local storage can be the "solution" for this. Local storage is a persistent storage method. Items stored in this location, do not have an expiry date and thus do not expire automatically. Additionally, Local storage can be accessed between different browser windows [44].

HTML5 offers another storage location, IndexedDB. In this database, JSON items are stored. A special feature of IndexedDB is the ability to add an index to items in the database. This way, certain entries can be found back more easily. IndexedDB is a persistant storage method, but is subjected to certain limitations. These are similar to HTML5 Local storage, being the same origin policy (which will be elaborated in section 3.1.) and only being able to read entries set by the same domain and protocol. For tracking purposes, HTML5 Local storage and IndexedDB have no real differences.

HTML5 Local storage and IndexedDB can be easily cleaned from within a web browser. Individual entries can be deleted trough the developer tools from popular web browsers, which can give an overview of storage used by web pages. HTML5 Local storage is included and entries can be deleted by the user. Another, far easier way to clear the Local storage is by using the "clear all cookies"-option of web browsers. This not only deletes all stored HTTP cookies, but also all HTML5 Local storage entries. This operation does not clear IndexedDB however. This will be further elaborated in section 3.2.

Usage in tracking

The characteristics mentioned above would make you think that HTML5 Local storage or IndexedDB could become an alternative to HTTP cookies. Currently, Local storage is used often, but together with HTTP cookies [5]. This means that Local storage is used for tracking, but has not replaced HTTP cookies entirely. These copies of HTTP cookies stored in HTML5 Local storage offer a way to respawn HTTP cookies.

For IndexedDB, the same goes as for HTML5 Local storage. They both offer yet another data storage location in which identifiers can be stored, but do not introduce any additional privacy threats with them, apart from cookie respawning (which nearly every storage space introduces). For IndexedDB specifically there is evidence that it is also used to rebuild Flash cookies [2]. Respawning Flash cookies instead of HTTP cookies is quite illogical, since HTTP cookies are sent automatically with HTTP requests. Flash cookies on the other hand must be obtained via JavaScript. Respawning Flash cookies instead of HTTP cookies can be done to give the tracker another fallback, if the user deletes both his HTTP and Flash cookies.

2.1.5 ETags

ETags are part of an HTTP response header. A webserver creates an ETag for its pages and sends this tag with the page when it is first requested by a browser. The browser will then send this ETag back in the `if-none-match` field when it requests the same page again. When the page is not altered, the server will reply this and the browser can just load the webpage from its cache instead of having to acquire it from the server again.

Usage in tracking

ETags are created by the webserver and does not have limitations other than the maximum size of 81864 bits. Because of this lack of restrictions, webservers can send unique tags to different users and thus track them, since the unique tag is always send back to the server when requesting the same page again. Ayenson *et al.* found that ETag tracking and cookie respawning with ETags was used by `kissmetrics.com` [5].

Tracking with ETags is hard to counter, since it relies on a core function of web browsers. To block tracking with ETags, you would have to clear the browser cache between each visit of a website.

2.2 Techniques using fingerprinting

Tracking can also be done without storing identifiers on the device of a user. The most common method for this is so-called fingerprinting. Here, a user, device or combination is identified by the information that they "leak" and "subtle but measurable variations which allow them to be fingerprinted."[18] This fingerprinting can be done in two ways: either active or passive. Passive fingerprinting is done by analysing information that is sent over the network in any case, without querying certain information. For instance, the IP address of a device is always seeable if this device connects to a web service. Active fingerprinting on the other hand actively queries the device for more information. This includes the operating system and the installed fonts of a device.

2.2.1 Passive fingerprinting

As told above, passive fingerprinting by a web service means observing the network traffic and from that, creating a unique identifier for the user/device combination. With this fingerprint, tracking parties can then follow users, even over longer periods of time. The properties of the user or device might change over time, but it has been proven by Peter Eckersley that it can be found out when a fingerprint is a "successor" of another fingerprint [18]. This could even be done with more than 99% accuracy in his case.

Passive fingerprinting can not be observed by users, since a web page is not sending any special requests or storing information on devices. All the fingerprinting and tracking happens server side. This makes it a really hard form of tracking to notice or counter and therefore an extremely interesting form of tracking for tracking parties.

2.2.2 Active fingerprinting

Active fingerprinting is more or less a tradeoff with passive fingerprinting, between being able to acces more information and thus being able to create a more unique fingerprint and users being able to spot and thus counter this fingerprinting. However, some counters to fingerprinting may even make it easier to fingerprint. Consider the example where Adobe Flash might give away certain details about your device. This makes it easier to create an accurate fingerprint. But when you disable Flash Player, this might make you part of an even smaller set of users, which then makes it even easier to fingerprint your device.

Active fingerprinting can be seen as an addition to passive fingerprinting. Most active techniques also make use of the information that is gathered with passive fingerprinting. Popular active approaches to fingerprinting are using the list of installed fonts [18] or using canvas fingerprinting [31]. In

this technique, the way a device handles drawing an image of some sort on a web page, is used as unique fingerprint. Very recently, Cao, Li and Wijmans found a fingerprinting method that can track users across different web browsers on a single device [13]. The method works by using hardware features of the device that is fingerprinted, such as the GPU and audio stack and is previewed at `uniquemachine.org`.

Every fingerprinting technique that requires an active request to a device, is seen as active fingerprinting.

Chapter 3

Privacy measures in web browsers

This chapter lists the most popular countermeasures against web tracking that are implemented in web browsers. Section 3.2 will cover all the configurable settings that are offered in browsers. Section 3.3 will focus on Private Browsing mode. Privacy enhancing browser extensions and ad blockers will be covered in Chapter 4. Although they are not really browser features, regulation and the Do Not Track and P3P incentives are also in this chapter (sections 3.1 and 3.2.2).

3.1 Regulation and policies

Since 2002, the European Union tries to regulate the placement and retrieval of information on devices of users by web pages. This was done through the ePrivacy directive [14], which ordered members of the EU to force web page owners to let users opt out of cookies (and other data storage). Only strictly necessary cookies were excluded. Mayer and Mitchell however concluded that most of the member states didn't enforce such laws and therefore the directive was somewhat of a failure [28].

In 2009 the opt-out of the directive was changed to an explicit opt-in [15]. In the Netherlands this change was adopted in the so-called "telecommunicatiewet" (article 11.7a [1]), commonly called the *cookie law*. This law required explicit consent from users and made web pages responsible for proving a certain cookie is not used for tracking purposes. The change in legislation led to web sites implementing different ways to aquire consent from the user. Amongst the solutions were so-called cookiewalls, which block the content of the website unless a visitor accepts all cookies, and implicit agreement when visiting a website. This "cookie war" between regulators and website owners (in the Netherlands) is nicely described by

[1] `http://wetten.overheid.nl/BWBR0009950/2016-11-03`

Ronald Leenes [24].

Other EU countries chose for a different approach. The UK and Spain for instance chose to require only implicit consent from users, but in Spain cookie placement is only legitimate if a user is active on a web page.[2]

3.2 Browser settings

The Popular browsers Google Chrome, Mozilla Firefox, Microsoft Internet Explorer/Egde and Safari[3] all offer user configurable settings which can benefit the privacy of users. The settings offered by web browsers boil down to:

- Settings on first and third party cookies
- Option to enable Do Not Track
- Option to enable Tracking Protection
- Settings on predictive services
- Settings on location services

An overview of how these settings are implemented in web browsers is given in table 3.1. Note that we have only looked into desktop versions of web browsers and not their mobile counterparts. For each setting, we will cover its goal, implementation in browsers and shortcomings in this section. Screenshots of the settings pages of the covered browsers can be found in Appendix A.

3.2.1 Settings on first and third party cookies

Goal

The goal of settings on cookies is to let users determine from which domains they want to accept cookies and from which ones they don't. Many cookies are used to store identifiers of users in, with which web sites can track their visitors, especially third party cookies. However, cookies are also used to keep useful states of web pages, such as a shopping basket on a web shop. Therefore, a good balance between these two kinds is needed.

Implementation

Settings on cookies are implemented in a similar way in most of the popular browsers. Chrome, Firefox and Edge let the user choose between allowing

[2] `https://cookiepedia.co.uk/cookie-laws-across-europe`
[3] `https://www.w3schools.com/browsers/`

all cookies, allowing no cookies and only allowing first party cookies. Google Chrome and Mozilla Firefox let the user also choose to store all cookies until the browsing session is closed. This means all cookies (also session cookies) are erased after closing the browsing session. Egde, Safari and Internet Explorer do not offer this option.

Safari by default blocks third party cookies, but it had a flaw which made it possible for a third party to still place cookies, even though the option of blocking them is turned on. It worked by submitting data through an HTML form. [27]. This bug was later fixed in WebKit, the browser engine used by Safari and adopted in Safari itself as well[4].

Internet Explorer handles the settings about cookies a bit differently. Different preferences can be set per network zone: internet, intranet, trusted web sites and restricted web sites. The options are embedded in a slider, which ranges from allowing all cookies, through blocking cookies that can personally identify a user or blocking cookies without a "compact privacy policy"(computer readable), to blocking all cookies. Internet Explorer uses the P3P[5] standard and uses this to determine which cookies are considered personally identifiable and which are not. P3P will be covered in section 3.2.2.

The highest privacy level in Internet Explorer also claims that cookies which are already on the computer can not be read by web pages. We found this to be true. Visiting a web page that uses cookies to keep users logged in, logging in on that page and then changing the privacy slider to the highest setting and refreshing the web page, no longer has the user logged in.

Limitations

The main limitation of disabling HTPP cookies, especially first party cookies, is that many web sites simply stop working, since they rely on cookies for their functionality. Another limitation of disabling (third party) HTTP cookies is that there are many alternatives available for trackers to still track you. Even with all cookies blocked, users can still easily get tracked. In my opinion, blocking first party cookies is no viable option, because of the mentioned functionality loss. Blocking third party cookies still is a viable option. If web sites do not work properly with third party cookies disabled and if users trust the site they are visiting, these sites can be added to an exeption list. This makes sure that third party cookies are only allowed on trusted domains.

[4] `https://support.apple.com/en-us/HT202425`
[5] `https://www.w3.org/P3P/`

3.2.2 Do Not Track and P3P

Goal

Other privacy increasing options are the Do Not Track (DNT) HTTP header field and the P3P standard. DNT was proposed in 2009 [41] and was standardized in 2015 [19], while P3P started in 2002 and was suspended in 2007.

DNT works by adding a field to HTTP request headers. The DNT header field can be either a `0`, `1` or `null`. These values mean that the user consents to being tracked, does not want to be tracked or has no preference respectively. A DNT-field of `1` should restrict web pages from setting tracking cookies or using other ways of tracking.

The P3P policy specifies which information about users is stored, how it is used and for how long. Users can also set a policy for themselves, which is compared with the server's policy. If the server wants to store more information than the users wants to, this is not allowed and the server will not set cookies with this unwanted information.

Implementation

Do Not Track is implemented the same in all the web browsers. It is an option which users can toggle. If Do Not Track is turned on, the browser will add a `DNT=1` field to the HTTP requests it makes. Firefox and Chrome tell the user what Do Not Track is and how it works. The other browsers do not.

P3P policies can be obtained as an XML file. Compact P3P policies can also be included in HTTP response headers from servers. P3P has only been active in Microsofts web browsers Internet Explorer and Edge. They offer certain levels of privacy in their settings (mentioned in section 3.2.1), which alters the P3P settings accordingly. In Edge and Internet Explorer 11 for Windows 10, the P3P compatibility was removed[6], because the functionality is seen as deprecated and because the standard was not adopted much.

Limitations

As of now, Do Not Track is just a policy with no effects at all, since the technique requires compliance from tracking parties, which will of course not easily comply to a standard that will constrain their business [38]. If implemented fully and adhered to by trackers, it could be a very promising technology. The general opinion however is that DNT will not work in its current form [2, 6, 11, 38, 22]. Radboud student Schileffski has done research creating requirements for for Do Not Track to work [39].

P3P has the same limitations as DNT, since P3P is also not enforced at all. This means that web pages do not need to have a P3P policy, and if they

[6] `https://msdn.microsoft.com/en-us/library/mt146424(v=vs.85).aspx`

have such a policy, they do not need to adhere to it. Moreover, P3P was not adopted much in 2002 [3], and still not in 2007 [9], when work on P3P was suspended. Also, sites that implemented P3P not always implemented it correctely [25] or not adhered to the policy at all [16].

3.2.3 Tracking Protection

Goal

Some browsers implement a function that is called Tracking Protection, which is a technique which is adopted from privacy extensions. This setting, if enabled, uses a list to block known trackers. With this option, browsers are trying to achieve a more private session for their users without the need of installing separate extensions.

Implementation

Mozilla Firefox and Microsoft Internet Explorer both offer Tracking Protection. In Firefox this will only work in private windows. This setting, if enabled, does not load content from known third party trackers based on a list of known trackers. Firefox uses a list from Disconnect[7](also covered in section 4) and offers users the possibility to change this list.

Limitations

Tracking Protection as implemented in Firefox and Internet Explorer has a few limitations. In Firefox, Tracking Protection can only be activated in private browsing mode. Also, users can only choose between the normal and strict blocking list offered by Disconnect, with no option to add or delete domains. Internet Explorer lets users pick any available blocking list, but this makes the user responsible for picking a good blocking list. Internet Explorer also does not offer the option to add exeptions to the block list.

3.2.4 Settings on predictive services

Goal

Google Chrome offers options to choose wether or not the browser may contact a web service that can predict search queries typed into the URL bar of the browser or that can present alternatives if the web page a user is trying to reach does not exist (navigation errors). It is important to note that this functionality only exists in Google Chrome, since it is the only web browser that makes use of predictive web services.

Disabling the webservices means that Google does not recieve the URL a user is trying to reach, with which Google can make a prediction.

[7] `https://disconnect.me`

Implementation

Google Chrome offers their users three choiches. They can separately enable or disable a web service for navigation errors, for completing URLs and search queries and a prediction service to load pages faster. The last option uses DNS to get the IP addresses of all links that are on a currently shown page. This saves the DNS lookup if a user navigates to a link that is embedded on the current page, but does not have any major privacy risks. The only stored information is DNS records of links on a page the user already visited. This information is also only stored until the browsing session is closed.

Limitations

Disabling the predictive services of Google Chrome has nearly no drawbacks for users. They might have to type more since their searches are not automatically completed or a they might stumble upon a 404: Not Found page, but the tradeoff is that Google does not recieve every entry in the URL bar a user enters.

3.2.5 Settings on location services

Goal

Safari, Chrome, Firefox, Edge and Internet Explorer offer the option to deny web pages acces to your physical location. The goal of this setting is to hinder advertisers in serving user location specific advertisements and to limit fingerprinting based on the location of devices and their users. The geolocation of a device is generated from a range of data sources, including the IP address and information about WiFi networks that a device is connected to. How WiFi networks can be used to determine the location of a device is explained by Kysela [23].

Implementation

The way that this setting is offered in browsers, is the same for Safari, Internet Explorer, Edge and Chrome. The user is offered the option to do not allow acces to the physical location of the device, which can be enabled or disabled. In Edge, if location services are allowed, the user is still asked for permission when he visits a web page that wants acces to location details for the first time.

Firefox has implemented this option differently. Each time a web page wants acces to the location of the user, he is promted with the option to allow this or not. If users want to disable location services entirely, meaning acces to location details will always be denied, they have to do this via the `about:config` file of Firefox.

Limitations

Disabling acces to location services has a big limitation for users, which is that determining the (rough) location of a device is still possible, for instance with the just IP address, which will always be "leaked". It might not be as accurate as with all the geolocation services enabled, but disabling these options do not make it impossible to find the location of your device. Therefore the option is, in my eyes, a bit misleading.

3.2.6 Summary of browser options

Looking at the different web browsers we covered in this research, we can compare them with regards to their privacy options. Mozilla Firefox and Google Chrome offer similar settings, with the difference between them being that Firefox offers Tracking Protection in private browsing mode, while Chrome has options on location and predictive web services. Firefox does not offer settings on predictive web services, because it has not implemented such services. Most of privacy settings Internet Explorer offers rely on P3P to work, which means that while the settings seem promising, they will not be effective in blocking web trackers. Therefore it is not strange that Edge has left out many of the settings that are available in Internet Explorer. Safari is the only browser of the ones we looked into which blocks third party cookies by default.

Browser:	**Firefox**	**Chrome**	**Edge**	**Internet Explorer**	**Safari**
Default cookie setting	Allow all	Allow all	Allow all	Allow all	Blocks third parties
Can block 3rd party cookies	Yes	Yes	Yes	Yes	Yes, but does not block all 3rd party cookies
Can add exeptions to cookie policy	Yes	Yes	No	Yes, but not possible if all cookies are blocked	No
Offers Do Not Track	Yes	Yes	Yes	Yes	Yes
Offers Tracking Protection	Only in private windows	No	No	Yes	No
Uses predictive services	Yes, from history and bookmarks	Yes, with web service	No	No	No
Can disable predictive services	Yes, separately	Yes, separately	Not applicable	Not applicable	Not applicable
Uses location services	Yes	Yes	Yes	Yes	Yes
Can disable location services	Yes, permission per site	Yes	Yes	Yes	Yes

Table 3.1: Privacy settings in popular browsers

Looking at the settings these web browsers offer with regards to privacy, we can see that Firefox and Chrome offer the most configurability. By default, Safari performs well because it blocks third party cookies with its default settings enabled. Another thing that comes forward is that Edge does not offer as much privacy settings as Internet Explorer does. This might be because users did not use the settings in Internet Explorer much and Microsoft wanted to focus on a simple overview of settings, without much configurability.

3.3 Private browsing mode

Private browsing mode (sometimes called incognito or InPrivate mode) is a privacy enhancing option that focusses on having a separate session from the "normal" browsing. The following section will look into the functionality of private browsing mode and the privacy increase it offers.

3.3.1 Different attacker models

Private browsing mode offfers protection against two different types of attacker or observer (from now called attacker). The first is a so-called local attacker, the second one is a remote attacker. A local attacker is an attacker that has physical acces to a user's device and tries to gather information through this. For instance, a co-worker or family member looking through your browsing history and by this, seeing what web sites you visited can be seen as a local attack. A remote attacker/observer on the other hand has no acces to the physical device of their target and tries to get information remotely. It is noteworthy that private browsing is mostly used to view adult content [12].

A remote attacker is more important in the scope of web tracking, since a web tracker essentially is a remote observer.

Goal

The target of private browsing mode is to obtain a more private session for users, agaist both a local and a remote observer. This is a difference with Tracking Protection (as covered in section 3.2.3) or Do Not Track, which only focus on a remote tracker. As can be seen in the Implementation section, all popular web browsers currently have the same focus, which is more focused on a local observer than a remote one.

Implementation

Private browsing mode is currently offered in all popular web browsers. Private browsing mode won't save browsing history, cookies or search history. This means that these three kinds of data are erased after the session is closed, instead of not kept at all, which would make web sites lose almost all their functionality. Private browsing offers an isolated session, which not only claims to be more private, but also can be used to be logged in to two accounts at once.

In Apple's Safari, browsing history, cookies and HTML5 Local storage states set in a "public" session were accesible in a private session in 2010 [12]. The other way around: viewing values set in a private session while in a public session, was not possible. However, the way history and cookie/HTML5 values set in a public session could be seen in a private session,

was abusable by web trackers. For instance, a user might have visited a website which used a cookie to keep track of this user. If the user then would like to have a separate session from the first visit, this would not be possible and a web tracker will easily connect the user in his private session to the same user in the earlier public session.

This shows that the private browsing mode in Safari was primarily focussed on protecting against local observers and not so much on offering protection against a remote tracker. This difference in attack model is something that should be kept in mind when looking into private browsing mode. Currently, as tested in in Safari version 10.0.3 on MacOS 10.12.3, the implementation of private mode is the same as in the other popular web browsers. When we opened a webpage that gives a new visitor a unique identifier in a cookie and used to keep users logged in, the cookie values in a new, *private* window, were different from the earlier set cookie in a public session and the user was no longer logged in. This means that cookies are no longer shared between public and private sessions.

Limitations

Private browsing mode is not without privacy risks. It stores cookies and browsing history in a separate location and removes these after browsing, but a local attacker might still see what is happening in the private session, since the information is still stored on the users computer. Remotely, other tracking techniques are still usable, such as fingerprinting the web browser.

Browser extensions and plugins might also leak information from a private session, which is not meant to be saved or accessible. Adobe Flash Player for instance, used storage on the user's computer which was not erased after quitting a private browsing session. Since Flash Player 10.1, this issue is fixed and Flash Player currently supports private browsing [32]. For Microsoft Silverlight there is no evidence that it violates or violated private browsing mode.

Bursztein *et al.* also show that there are numerous other, sometimes less popular, browser extensions which might pose a threat to private browsing mode [12]. These include the popular extension NoScript.

Chapter 4

Browser extensions for privacy

This chapter presents the techniques that are applied by popular privacy enhancing browser extensions. Section 4.1 will cover the techniques, their implementation in extensions and their shortcomings. Section 4.2 will give an overview of the techniques and compares extensions with each other. Section 4.3 will present a recommended set-up for users to be more private while not losing usability.

4.1 Privacy enhancing browser extensions

Since the early 2000s[1], web browsers support so called extensions or plug-ins. These are user-created pieces of software that run inside a web browser. Two kinds of browser extensions are interesting to look at from a privacy perspective: privacy enhancing extensions and advertisment blocking extensions. These extensions are worth looking into because they both try to enhance the privacy of users, directly or indirectly. In the following sections, the techniques applied by the most popular privacy enhancing browser extensions(determined by the most downloads from Mozilla Firefox and Google Chrome's extension stores) will be elaborated. The covered extensions are: Disconnect, Privacy Badger, Ghostery, AnonymoX, AdBlock Plus and uBlock Origin. The applied techniques include:

- Blocking known trackers with a blacklist
- Identifying trackers with algorithms
- Replacing social network buttons
- Other services, such as offering a proxy server or restricting acces to the canvas

[1] Internet Explorer: 1999, Firefox: 2004, Chrome: 2010

4.1.1 Blocking known trackers with a blacklist

Goal

The goal of blocking trackers that are on a blacklist is straightforward. All requests to domains that are classified as a tracker are blocked. Many blacklists exist and extensions can choose to incorporate them, or create their own blacklist.

Implementation

Most of the extensions we looked into, make use of blacklists to block requests to known trackers or advertisers. The blacklist Disconnect uses, currently consists of over 2000 third parties, although many of these are subdomains of larger parties. Disconnect offers users an option to whitelist (first party) sites, which will then be able to make requests to third party web sites. This means that you can for instance whitelist `ru.nl`, which allows `ru.nl` to make requests to any third party tracker. Although this concept blocks a large portion of trackers already, Disconnect by default enables the requests of *content* third parties. The difference between content and tracking third parties is rather small, but can be seen in the following way: if an embedded third party is "useful" to the web page, it is seen as content. This is for instance the case if a website embeds Google Maps elements on their contact page. Because the difference between content and tracking parties is so small, Disconnect has chosen to work with a whitelist of third party content providers, which consists of 110 domains currently. Domains on this list are seen as pure content providers and thus their requests are allowed.

Ghostery is not open source, but claims to have the largest tracker database of all the privacy tools offered. This can however not be easily verified, since the list is not publicly available.

AdBlock Plus works with a list of known domains to which it will block requests. Of course, the lists AdBlock Plus uses by default and the extra lists that can be added (via `https://easylist.to/`) focus on advertisements primarily instead of pure web trackers.

uBlock Origin acts in the same way as AdBlock plus does, by using blocklists to block advertisements. uBlock Origin itself claims it is not just an ad blocker, but rather a "wide-spectrum" blocker. By this they mean that the uBlock Origin extension can also be set up to block third parties altogether or to block scripts embedded on web pages.

Limitations

Using blacklists with trackers to block requests has some drawbacks. Lists can lack trackers, or have domains on them, which are not trackers. Also, extensions might choose to incorporate lists (by default) that do not have

most of the known tracking domains on them. This leaves it up to users to add better lists, if this possibiliy is offered by the extension.

Another limitation of blacklists is that domains that not only track, but also serve content, might be blocked because they are on a blacklist. In this situation, you would have to choose between allowing the tracking and having the functionality or losing the functionality but also blocking the tracking.

4.1.2 Identifying trackers with algorithms

Goal

The goal of identifying trackers with algorithms and then blocking them is similar to blocking known trackers with blacklists. Every domain that acts as a tracker, as determined by an algorithm, is blocked. The advantage over blacklists is that as long as the algorithm works correctely, trackers are blocked. There is no blacklist that needs to be kept up to date.

Implementation

Privacy Badger relies on algorithmic blocking for its functionality. The algorithm applied by Privacy Badger keeps track of third party domains that a user visits (as presented by Bau *et al.* [8]). If this domain appears to be using identifying cookies, super cookies in local storages or requests acces to the canvas (as presented by Acar *et al.* [2]), it is registered as a tracker. When a domain also serves functional content, the algorithm filters out the tracking parts and allows the content.

Ghostery, which is acquired by Cliqz (covered in section 4.2), will also be offering algorithmic blocking in the future. The algorithm (presented in their paper from 2016 [47]),already used in the tool from Cliqz, will also be integrated into Ghostery according to their blogpost[2] on the take-over.

Limitations

Algorithmic blocking of trackers has some limitations, which are similar to the limitations of blocking trackers with blacklists. Users have to rely on the algorithm to identify trackers correctly and also, to identify non-trackers as such. If the algorithm blocks functional content, exceptions can be added but it of course lowers usability if users will have to add a lot of exceptions to have working web sites. It might also be possible that an algorithm does not identify trackers as trackers and therefore allows them, which should not happen.

[2]Available on `https://www.ghostery.com/blog/ghostery-news/ghostery-acquired-cliqz/`.

4.1.3 Replacing social network buttons

Goal

Ghostery and Privacy Badger choose to handle social network buttons such as the Facebook "Like" button and Google's "+1" differently than other elements from those domains. They do this because many users still want to use the functionality from the buttons, but not want to share their data with Facebook or Google on each page that has a social button embedded. This is done because many social networks that have such buttons, use these buttons to track users. When a website embeds a Facebook button for instance, the user's Facebook identifier along with the URL on which the button is embedded, is sent to Facebook[3].

Implementation

To block the social buttons, the EFF has incorporated ShareMeNot (no longer active since the incorporation in 2014) into Privacy Badger. It works by blocking the requests to the social networks behind the buttons at first, but still rendering the buttons. Only when a users clicks on a social button, requests are sent to the social network behind the button.

Ghostery, like Privacy Badger, also limits the functonality of social media buttons and works the same way.

Limitations

Replacing social network buttons does not have any limitations in my eyes. The only limitation it has is for the user, who has to click twice on the button instead of once if he wants to use the button.

4.1.4 Other techniques

Goal

Other techniques that browser extensions apply to limit web tracking are offering VPNs and proxy servers and altering the Geo-ID of the web browser. The main goal of this is to make a browser/device combination less unique and therefore harder to fingerprint. This then makes it harder for web trackers to keep track of these devices and their users across the web.

Not allowing acces to the browser canvas, or only allowing acces after the user has allowed this, makes it impossible for web pages to use the (rendering of images on) the canvas in order to fingerprint devices. This then makes it harder to track users and devices, since their fingerprint is less specific.

[3] `https://www.facebook.com/help/186325668085084`

Implementation

Disconnect offers a VPN only in their premium subscription. AnonymoX offers different anonymization functions. These functions are: changing your IP address to one provided by AnonymoX through a proxy server, altering your Geo-ID and clearing cookies from certain websites. For altering the IDs, AnonymoX uses a anonymization network, consisting of servers in every country in a user's browser country list.

Limitations

The main drawback of VPNs and proxy servers is that they have to be trusted. It is of no use if users send their traffic through a proxy to be more anonymous, while the proxy still tracks of all their users. Another drawback is that the speed with which web pages are loaded is reduced severely, which of course limits the usability.

4.2 Summary of privacy enhancing extensions

4.2.1 Business models

An important aspect of browser extensions to keep in mind when comparing them is their business model. A non-profit organisation that brings out an extension will most likely have other goals in mind than a commercial company that also makes money from an advertisement network.

Disconnect makes money through the sales of their premium options, which are standalone applications additional to the browser extension. These applications act as anti virus / anti malware software. They also offer a VPN in their premium subscription.

There have been claims that Evidon (previously Ghostery inc.) makes money from selling user information[4] to advertisement companies [40, 10]. Ghostery itself is not clear in how it collects and monetizes this data and claims that most users choose not to share data with Ghostery. Sharing "page and tracker" data is an option in the extension that can be enabled or disabled, but the information about the data and for what it is used is minimal.

In 2010, Ghostery temporarily made their source code open source[5], but currently it is a closed-source proprietary extension.

As of february 15th, 2017, Ghostery is part of *Cliqz*, a German company that is owned by Hubert Burda Media and Mozilla. Ghostery anounced to continue working closely with Evidon as well in their blogpost[6] on the take-

[4] Ghostery gives an overview of the collected data on `https://www.ghostery.com/faq`.

[5] Available on `https://github.com/jonpierce/ghostery`.

[6] Available on `https://www.ghostery.com/blog/ghostery-news/ghostery-acquired-cliqz/`.

over. They also state that nothing about the aforementioned data collection will be changed.

The basic version of AnonymoX does not block advertisements. The premium version, with which they earn their money, also includes an ad blocker.

AdBlock Plus is open source[7] and currently part of the German Eyeo GmbH.

In 2011, the "Acceptable Ads" list was added, which caused a lot of controversy . This list is a list of advertisement domains considered "not intrusive" by AdBlock Plus. The ABP extension does offer the option to not allow Acceptable Ads, but they are allowed by default. Pujol *et al.* have found out that many users who use AdBlock Plus have Acceptable Ads turned on (and no other than the default blocking list activated), since this is the default setting. [36]. There have been claims that AdBlock Plus is having ties with advertisement companies as a result of incorporating Acceptable Ads [4, 35, 20, 30].

In 2015, Walls *et al.* researched the Acceptable Ads program and concluded that disclosures of the financial relationships between AdBlock plus and advertisement networks and an open discussion about it are neccesary to keep the trust of users and to reach an agreement between users and advertisement networks [45].

The uBlock Origin project, which is run by multiple people, is owned by Raymond Hill. He is the founder and original developer of *uBlock*, which currently goes by the name of uBlock Origin. The project refuses donations of any kind and is not monetized in another way.

Since Privacy Badger is developed by the EFF, which is a not-for-profit foundation, their business model is quite clear. The foundation relies on donations to cover it expenses, but does not monetize its work.

4.2.2 Discussion

All the mentioned countermeasures against web tracking are, to a certain extent, effective. The functionality of the privacy tools and ad blockers is similar, but the extensions all have their own focus. Ad blockers mainly focus on blocking advertisements, but AdBlock Plus focusses mainly on blocking (in their eyes) intrusive advertisements, whereas uBlock Origin tries to block all advertisements, as well as increase the privacy of its users. This is where the main focus of the privacy extensions is. They might also block advertisements, but only if they are listed as a tracker.

Another diffence between the tools is the business model they adhere to. Privacy Badger and uBlock Origin are non-profit tools, while Ghostery

[7] Available on `https://github.com/adblockplus/adblockplus`.

and AdBlock Plus have a much more commercial business model in which processing or selling user data is no exeption.

The main method of blocking trackers / advertisements of the tools is the same: blocking requests to domains that are registered or behave as trackers. This means that, as long as a good blocking list is used or the algorithm to determine what is a tracker is sound, the tool will keep most of data-storage based tracking away. One of the tracking methods that is not countered by the mentioned extensions is tracking with ETags. The extensions do not clear the browser cache between web page visits.

Another tracking method that the covered extensions do not (fully) counter is fingerprinting. Of course, the main reason for this is that fingerprinting is very hard to spot and to counter. Especially passive fingerprinting, fingerprinting based on information that a device will always "leak", is almost incounterable. The only extension that tries to counter this is AnonymoX, with their IP and Geo ID changing functionality. This doesn't make it unable for websites to fingerprint you, but will give a fingerprint that is somewhat random and not directely connected to your device.

Active fingerprinting is somewhat easier to detect, since an active query of some sort (for instance rendering something on the canvas) is sent to the device. The hard part however, is determining which activities are used to create a fingerprint and which are used for legitimate purposes. Privacy Badger can disable canvas fingerprinting, but also states that countering other types of fingerprinting are ongoing projects. Of course, it is somewhat of a race between organisations like the EFF finding a counter to fingerprinting techniques and web trackers finding a new way of fingerprinting.

As long as fingerprinting is not effectively countered by popular privacy-enhancing tools, users of such a tool who think it makes it unable to follow them around the web, are wrong. Of course, the functionality that the current popular tools have, does have a serious impact on lowering the tracking possibilities via data storage, but web trackers also have this knowledge. Because of this, they might shift more towards fingerprinting which is much harder to counter. For trackers, a lot of potential income is at stake [34], so having an as persistant as possible tracking method is in their economic advantage.

There has also been research by others into the field of ad blockers and privacy tools. Wills and Uzunoglu concluded that tools that can not be configured such as Disconnect do not block much of the third party tracking domains [46]. They found out that uBlock (Origin) performs best and AdBlock Plus only provides enough protection if blocking lists are manually added. Ghostery does not provide any protection by default in their eyes, since blocking of trackers must be turned on by the user. Wills and Uzunoglu did not look into Privacy Badger however.

Doruk Uzunoglu also wrote his PhD thesis about ad blockers, in which he presents an overview of ad blockers from a user's perspective, popular third parties and blocking lists [43]. He shows that a majority of the third parties are in the *AdTrackers* category, which means that they are third parties that serve advertisements and track the behaviour of users across the web. Another thing Uzunoglu shows, is that the blocking list hpHosts has the highest blocking rate, of 91%. The Acceptable Ads list allows 29% of the most popular domains according to Uzunogly.

Both these researches have only focussed on ad blockers, Tracking Protection (as covered in section 3.2.3) and Ghostery, but not on other privacy tools or other functionality web browsers offer to counter web tracking.

Privacy tool:	**Disconnect**	**Privacy Badger**	**Ghostery**	**AnonymoX**	**AdBlock Plus**	**uBlock Origin**
Active since	2011	2014	2008	2010	2006	2014
Open Source	Yes	Yes	No	No	Yes	Yes
Business model	Sells premium versions	Not-for-profit	Sells user data	Sells premium versions	Not clear	Does not make or get money
Uses blacklists to block trackers	Yes	No	Yes, but will change to algorithms	Yes	Yes	Yes
Replaces social buttons	No	Yes	Yes	No	No	No
Uses algorithm to block trackers	No	Yes	No, but will in the future	No	No	No
Offers VPN	Yes, in premium version	No	No	No	No	No
Offers Proxy service	No	No	No	Yes	No	No
Alters Geo-ID	No	No	No	Yes	No	No

Table 4.1: Overview of popular privacy tools and ad blockers

In this overview above we see a clear disctinction between extensions that use blacklists to block trackers and extensions that use algorithms for this. I personally think that algorithmic blocking of trackers will be used increasingly more by browser extensions. Identifying trackers with an algorithm is much more dynamic than using blacklists, which have to be updated constantly. The downside is of course that designing a tracker identifying algorithm is a lot harder than maintaining a list with tracker domains on them.

4.3 User recommendation

So, after looking into the most popular tracking protection extensions available and the functionality web browsers offer against web tracking, what would be a good configuration for users to minimize web tracking, but not lose usability?

The recommended choice for a web browser is Mozilla Firefox. The reason for this choice is that it is an open-source web browser, which is not owned by a purely commercial party. Google and Microsoft are not only commercial companies, but also have their own advertisement networks. Internet Explorers privacy settings looked promising when P3P was new, but most of the settings are deprecated as of now. This means that the only viable settings Internet Explorer has is to block all cookies or to not block any cookies. Safari is only a viable choice for Mac-systems, because

development for other operating systems stopped. For Mac OS, Firefox is also available, which we recommend.

The main advantages of Firefox on top of the availability for many platforms, are the large amount of available extensions and the configurability of the browser.

Finetuning the settings of the browser is the next step. For Mozilla Firefox to be as private as possible, certain settings need to be enabled.

- Block pop-up windows, to prevent first party cookies from them
- Use tracking protection in private windows
- Change the block list to Disconnect's strict list
- Enable Do Not Track
- Use custom settings for history

  Keep cookies until the session is ended

  Never accept third party cookies

As for browser extensions, really only two extensions are recommended. The first is Privacy Badger, which is non-commercial. Inside Privacy Badger, users should enable the options to replace social widgets and to prevent WebRTC from leaking the local IP address (not covered, see Chapter 5).

The second recommended extension is uBlock Origin, with at least the EasyPrivacy list enabled. uBlock Origin is prefered over AdBlock Plus because uBlock Origin does not allow acceptable ads and has a much clearer business model.

The combination of these two extensions will try to minimize being tracked on the web, while also removing unwanted advertising. uBlock origin will block many trackers by default[8] and the algorithmic blocking of Privacy Badger has the potential to block remaining tracking domains. The only recommendation left is to always disable plugins like Adobe Flash and Microsoft Silverlight and only enable them on trusted domains that cannot do without.

This recommendation will not have a large impact on usability. Configuring the settings in Firefox is very straightforward and is a one-time operation. Installing Privacy Badger and uBlock Origin is exactly the same as installing any other extension for Firefox. Enabling the replacing of social buttons and prevention of WebRTC leaking the local IP address can be done via a simple checkbox in the settings of Privacy Badger. For uBlock Origin, the same

[8] `https://github.com/gorhill/uBlock/wiki/uBlock-and-others%3A-Blocking-ads%2C-trackers%2C-malwares`

goes; the EasyPrivacy list is enabled through a checkbox in the options of the extension.

As for usability of web sites when a user has configured his browser as in this recommendation, there should be almost no loss of functionality. Only, since the cookies are deleted after each session, users can not stay logged in on web pages for longer than the session. If web pages do not function correctly, users can determine to add an exeption for the page. This would then have to be done in Firefox as well as the installed extensions. This is also a reason why we do not choose to recommend installing all mentioned extensions. Adding an exception would then have to be done in all extensions, which is not good for usability. Additionally, since most extensions can all have the same blacklists added to them, functionality would not increase if more extensions are installed.

The Dutch digital rights organisation Bits Of Freedom (BOF) also presents a user recommendation on their website[9]. This recommendation covers which web browser is recommended and presents an overview of possible privacy tools to use. It does not go into the techniques that are used by browsers and extensions however.

BOF recommends Mozilla Firefox as a web browser, because of its customizability and because Chrome has some shortcomings in their opinion, such as that Google is not clear in what it uses synchronized data from Chrome for. BOF does not cover Microsoft Internet Explorer/Edge or Safari on their website. From our research, we can conclude that these web browsers do not perform better for privacy when configured correctely. Safari does perform better by default however, since it blocks third party cookies in this default.

From the tools covered in this research, Privacy Badger, uBlock Origin, Ghostery and Disconnect are also present on the BOF website. Privacy Bagder and uBlock Origin are considered good privacy tools by BOF, but Ghostery and Disconnect might not be privacy-friendly. This has to do with their business model. AdBlock Plus and AnonymoX are not covered by BOF. We can conclude that AdBlock Plus and AnonymoX do not perform better or have a clearer business model than the extensions both we and BOF recommend.

[9] `https://toolbox.bof.nl/playlist/prive-online/`

Chapter 5

Future Work

This thesis looked into popular web tracking methods and popular countermeasures offered in web browsers and browser extensions. While conducting this research, some things were left out and some other research topics came into mind, but were not covered in this research.

With mobile applications more and more taking over the traditional desktop applications, it is interesting to see how mobile web browsers handle web tracking and fingerprinting. Additionally, there are advertisement blocking and privacy-enhancing applications on offer. Checking which tracking methods these apps are effective against, could be an interesting research topic. The same goes for different desktop operating systems.

The Tor web browser is also an interesting topic, since it claims to have a unique fingerprint and counters to most tracking techniques. Verifying whether this is the case and if no tracking options exist that might still be able to track users using the Tor browser is very interesting.

We did not look into one functionality of Privacy Badger, preventing WebRTC from leaking the local IP address[1]. This privacy measure can be looked into in future research and comparison of privacy enhancing browser extensions.

The overview of tracking mechanisms and countermeasures can always be extended with less popular options or new, upcoming tools. Some options might even fall out of the list given in this thesis, when it is no longer supported or has fallen in popularity. This might happen for Microsoft Silverlight. It might be interesting to determine what tracking options are used by trackers that formerly used Silverlight (for instance *evercookie*[2], although this uses other storage locations as well).

In the scope of ad blocking and their blocking lists, trying to create the most optimal blocking list might be interesting. Another option is to try and

[1] Explained at `https://threatpost.com/webrtc-found-leaking-local-ip-addresses/110803/`.

[2] `http://samy.pl/evercookie/`

optimize the algorithms put to use by browser extensions such as Privacy Badger. This will hopefully give a better way to determine what party is and what is not a web tracker.

One of the most interesting fields of research in this area is finding new ways in which known tracking methods that currently do not have a countermeasure, are counterable. Finding countermeasures to fingerprinting techniques are still very much ongoing research.

In this research, we have mentioned limitations from web browsers and privacy enhancing extensions. In a future research project, these limitations might be fixed or a recommendation about which techniques to apply and which defaults should be set can be written.

Chapter 6

Conclusions

Conclusions on web tracking

From the overview of popular web tracking methods and popular countermeasures, several conclusions can be drawn. Cookie respawning can be achieved through every storage option web browsers and plugins offer. The Do Not Track and P3P incentives will not work, because of the required compliance of tracking parties. There are no countermeasures against passive fingerprinting, since passive fingerprinting relies on information that is always sent by a device while browsing the web. More and more web tracking and fingerprinting techniques are being developed which can track users ever more precisely. On the other hand, countermeasures are also evolving to cover these tracking methods. The cat-and-mouse race is still very much ongoing.

Conclusions on browsers

When looking at the differences between the most popular web browsers, it became clear that the web browsers currently all offer similar privacy settings. Of the covered browsers, Mozilla Firefox offers the best configurability. The defaults of these settings are also not really different, apart from Safari, which blocks third party cookies by default. This is in my eyes a very good default setting, which already blocks quite some trackers.

Private browsing mode has a focus that is different than other techniques that improve privacy. Private browing focusses more on a local attacker than on a remote one. It is important to keep the difference between a local and remote attacker or observer in mind. The implementation of private browsing is currently implemented similar in all the browsers. Safari previously had a different implementation that had some flaws, but has fixed this in the meantime.

As far as the configurability of web browsers goes, Mozilla Firefox and

Google Chrome offer the most privacy settings for their users. Internet Explorer has a lot of privacy levels the user can choose from, but since they nearly all rely on P3P to work, they are of no real use. Another advantage of Firefox and Chrome is the large amount of extensions that are available.

Conclusions on privacy tools and ad blockers

The privacy tools I looked into in this thesis, almost all make use of the same technique using blacklists with known trackers in order to block requests to them. Currently, only Privacy Badger uses algorithmic blocking. Blacklisting can be an effective technique, but for this it is important to choose a good list with good coverage of trackers.

In my eyes algorithmic blocking has more potential than blacklists, because algorithmic blocking is much more dynamic than using blacklists and removes the need to constantly update a blacklist. Also the large amount of blacklists that are currently used by numerous tools can be replaced by an algorithm with different levels of protection, making it easier for users to configure privacy tools.

Some privacy enhancing extensions apply some additional techniques, such as replacing social widgets or offering a proxy server.

With browser extensions, it is important to keep the business model of the author of the extension in mind. Some extensions are owned and created by commercial companies that are also advertisement companies or have ties to them. Therefore and because of their functionality and high usability, my personal recommendation is a combination of Privacy Badger, which is created by the EFF and not for profit uBlock Origin.

Reflection

Looking back on the process of this thesis, I can conclude that the field of web tracking is a very large, broad and rapidly changing field. As a result of this, almost every aspect of web tracking and privacy related concerns because of this can be a research subject. The hardest part I encountered in the process of doing research, was limiting the scope to something insteresting to research, yet feasible for a bachelor thesis. First, I looked into tools that analyse online advertisements and into cookie respawning. These were not continued because of the tools not working properly and cookie respawning being too small of a subject with my chosen approach. After some time, I decided for the current approach which focusses on tracking techniques, browser settings and browser extensions. Most other research done in this area chooses to only focus on one aspect of these three. Because of this, it is sometimes not clear what countermeasures to tracking are implemented and actually in use by web browsers or extensions.

Bibliography

[1] M. Abraham, Cameron Meierhoefer, and Andrew Lipsman. The impact of cookie deletion on the accuracy of site-server and ad-server metrics: An empirical comscore study. *Retrieved October*, 14:2009, 2007.

[2] Gunes Acar, Christian Eubank, Steven Englehardt, Marc Juarez, Arvind Narayanan, and Claudia Diaz. The web never forgets: Persistent tracking mechanisms in the wild. In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*, pages 674–689. ACM, 2014.

[3] Annie I Antón, Julia Brande Earp, and Angela Reese. Analyzing website privacy requirements using a privacy goal taxonomy. In *Requirements Engineering, 2002. Proceedings. IEEE Joint International Conference on*, pages 23–31. IEEE, 2002.

[4] Unknown Author. Serious accusations against AdBlock Plus, 2013. `https://web.archive.org/web/20131208011244/http://www.h-online.com/newsticker/news/item/Serious-accusations-against-AdBlock-Plus-1897360.html`.

[5] Mika D. Ayenson, Dietrich James Wambach, Ashkan Soltani, Nathan Good, and Chris Jay Hoofnagle. Flash cookies and privacy ii: Now with html5 and etag respawning. *Available at SSRN 1898390*, 2011.

[6] Rebecca Balebako, Pedro Leon, Richard Shay, Blase Ur, Yang Wang, and L Cranor. Measuring the effectiveness of privacy tools for limiting behavioral advertising. Web, 2012.

[7] A. Barth. RFC 6265: Http state management mechanism. `https://tools.ietf.org/html/rfc6265`.

[8] Jason Bau, Jonathan Mayer, Hristo Paskov, and John C Mitchell. A promising direction for web tracking countermeasures. *Proceedings of W2SP*, 2013.

[9] Patricia Beatty, Ian Reay, Scott Dick, and James Miller. P3p adoption on e-commerce web sites: a survey and analysis. *IEEE Internet Computing*, 11(2), 2007.

[10] Ricardo Bilton. Ghostery: A web tracking blocker that actually helps the ad industry, 2012. Available: http://venturebeat.com/2012/07/31/ghostery-a-web-tracking-blocker-that-actually-helps-the-ad-industry/.

[11] Tomasz Bujlow, Valentín Carela-Español, Josep Solé-Pareta, and Pere Barlet-Ros. Web tracking: Mechanisms, implications, and defenses. *arXiv preprint arXiv:1507.07872*, 2015.

[12] Gaurav Aggarwal Elie Bursztein, Collin Jackson, and Dan Boneh. An analysis of private browsing modes in modern browsers. In *Proceedings of the 19th USENIX Security Symposium*, 2010.

[13] Yinzhi Cao, Song Li, and Erik Wijmans. (cross-)browser fingerprinting via os and hardware level features. *Lehigh University*, 2017.

[14] European Commission. Directive on privacy and electronic communications, 2002. http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=CELEX:32002L0058:en:HTML.

[15] European Commission. Directive 2009/136/ec of the european parliament and of the council of 25 november 2009, 2009. http://eur-lex.europa.eu/legal-content/EN/TXT/HTML/?uri=CELEX:32009L0136&from=EN.

[16] Lorrie Faith Cranor. Internet explorer privacy protections also being circumvented by google, facebook, and many more. *Techpolicy.com*, 2012. Available on http://www.techpolicy.com/Cranor_InternetExplorerPrivacyProtectionsBeingCircumvented-by-Google.aspx.

[17] Ivar Derksen. HTML5 Tracking Techniques In Practice, 2016. Bachelor Thesis. Radboud University Nijmegen.

[18] Peter Eckersley. How unique is your web browser? In *International Symposium on Privacy Enhancing Technologies Symposium*, pages 1–18. Springer, 2010.

[19] Roy T. Fiedling and David Singer. Tracking preference expression (DNT). https://www.w3.org/TR/tracking-dnt/.

[20] Nermin Hajdarbegovic. Adblock plus denies ad fixing allegations, 2013. http://www.techeye.net/business/adblock-plus-denies-ad-fixing-allegations.

[21] Chris Jay Hoofnagle, Ashkan Soltani, Nathan Good, Dietrich James Wambach, and Mika D.s Ayenson. Behavioral advertising: the offer you cannot refuse. *Harvard Law & Policy Review*, pages 273–96, 2012.

[22] Georgios Kontaxis and Monica Chew. Tracking protection in firefox for privacy and performance. *arXiv preprint arXiv:1506.04104*, 2015.

[23] Jiří Kysela. Comparison of web applications geolocation services. In *Computational Intelligence and Informatics (CINTI), 2014 IEEE 15th International Symposium on*, pages 449–453. IEEE, 2014.

[24] Ronald Leenes. The cookiewars: From regulatory failure to user empowerment? *The Privacy & Identity Lab*, 2015.

[25] Pedro Giovanni Leon, Lorrie Faith Cranor, Aleecia M McDonald, and Robert McGuire. Token attempt: the misrepresentation of website privacy policies through the misuse of p3p compact policy tokens. In *Proceedings of the 9th annual ACM workshop on Privacy in the electronic society*, pages 93–104. ACM, 2010.

[26] Tai-Ching Li, Huy Hang, Michalis Faloutsos, and Petros Efstathopoulos. Trackadvisor: Taking back browsing privacy from third-party trackers. In *International Conference on Passive and Active Network Measurement*, pages 277–289. Springer, 2015.

[27] Jonathan Mayer. Web policy. safari trackers.[online] web policy blog, february 17, 2012. Available on `http://webpolicy.org/2012/02/17/safari-trackers/`.

[28] Jonathan R. Mayer and John C. Mitchell. Third-party web tracking: Policy and technology. In *2012 IEEE Symposium on Security and Privacy*, pages 413–427. IEEE, 2012.

[29] Aleecia M. McDonald and Lorrie Faith Cranor. Survey of the use of adobe flash local shared objects to respawn http cookies, a. *ISJLP*, 7:639, 2011.

[30] Mollu McHugh. Media mafiosos: is adblock plus shaking down websites for cash to let ads through?, 2013. `http://www.digitaltrends.com/web/adblock-plus-accused-of-shaking-down-websites/`.

[31] Keaton Mowery and Hovav Shacham. Pixel perfect: Fingerprinting canvas in html5. *Proceedings of W2SP*, 2012.

[32] Jimson Xu & Tom Nguyen. Private browsing in flash player 10.1, June 2010. `http://www.adobe.com/devnet/flashplayer/articles/privacy_mode_fp10_1.html`.

[33] PageFair. The 2015 ad blocking report. `https://blog.pagefair.com/2015/ad-blocking-report/`.

[34] PageFair and Adobe. The cost of ad blocking, 2015. https://downloads.pagefair.com/wp-content/uploads/2016/05/2015_report_the_cost_of_ad_blocking.pdf.

[35] Sascha Pallenberg. AdBlock Plus undercover – einblicke in ein mafioeses werbenetzwerk, 2013. https://www.mobilegeeks.de/adblock-plus-undercover-einblicke-in-ein-mafioeses-werbenetzwerk/.

[36] Enric Pujol, Oliver Hohlfeld, and Anja Feldmann. Annoyed users: Ads and ad-block usage in the wild. In *Proceedings of the 2015 ACM Conference on Internet Measurement Conference*, pages 93–106. ACM, 2015.

[37] J. Reschke R. Fielding. RFC 7230: Hypertext transfer protocol (http/1.1): Message syntax and routing. https://tools.ietf.org/html/rfc7230.

[38] Franziska Roesner, Tadayoshi Kohno, and David Wetherall. Detecting and defending against third-party tracking on the web. In *Proceedings of the 9th USENIX conference on Networked Systems Design and Implementation*, pages 12–12. USENIX Association, 2012.

[39] Patrick Schileffski. Do not track - wat het is en wat het zou moeten zijn, 2012. Bachelor Thesis. Radboud University Nijmegen.

[40] Tom Simonite. A popular ad blocker also helps the ad industry. *MIT Technology Review*, 2013. Available: https://www.technologyreview.com/s/516156/a-popular-ad-blocker-also-helps-the-ad-industry/.

[41] Christopher Soghoian. The history of the do not track header. http://paranoia.dubfire.net/2011/01/history-of-do-not-track-header.html.

[42] Ashkan Soltani, Shannon Canty, Quentin Mayo, Lauren Thomas, and Chris Jay Hoofnagle. Flash cookies and privacy. In *AAAI spring symposium: intelligent information privacy management*, volume 2010, pages 158–163, 2010.

[43] Doruk Uzunoglu. *Understanding ad blockers*. PhD thesis, Worcester Polytechnic Institute, 2016.

[44] Patrick Verleg. Cache Cookies: searching for hidden browser storage, 2014. Bachelor Thesis. Radboud University Nijmegen.

[45] Robert J Walls, Eric D Kilmer, Nathaniel Lageman, and Patrick D McDaniel. Measuring the impact and perception of acceptable advertisements. In *Proceedings of the 2015 ACM Conference on Internet Measurement Conference*, pages 107–120. ACM, 2015.

[46] Craig E Wills and Doruk C Uzunoglu. What ad blockers are (and are not) doing. In *Hot Topics in Web Systems and Technologies (HotWeb), 2016 Fourth IEEE Workshop on*, pages 72–77. IEEE, 2016.

[47] Zhonghao Yu, Sam Macbeth, Konark Modi, and Josep M Pujol. Tracking the trackers. In *Proceedings of the 25th International Conference on World Wide Web*, pages 121–132. International World Wide Web Conferences Steering Committee, 2016.

Appendix A

Browser privacy settings

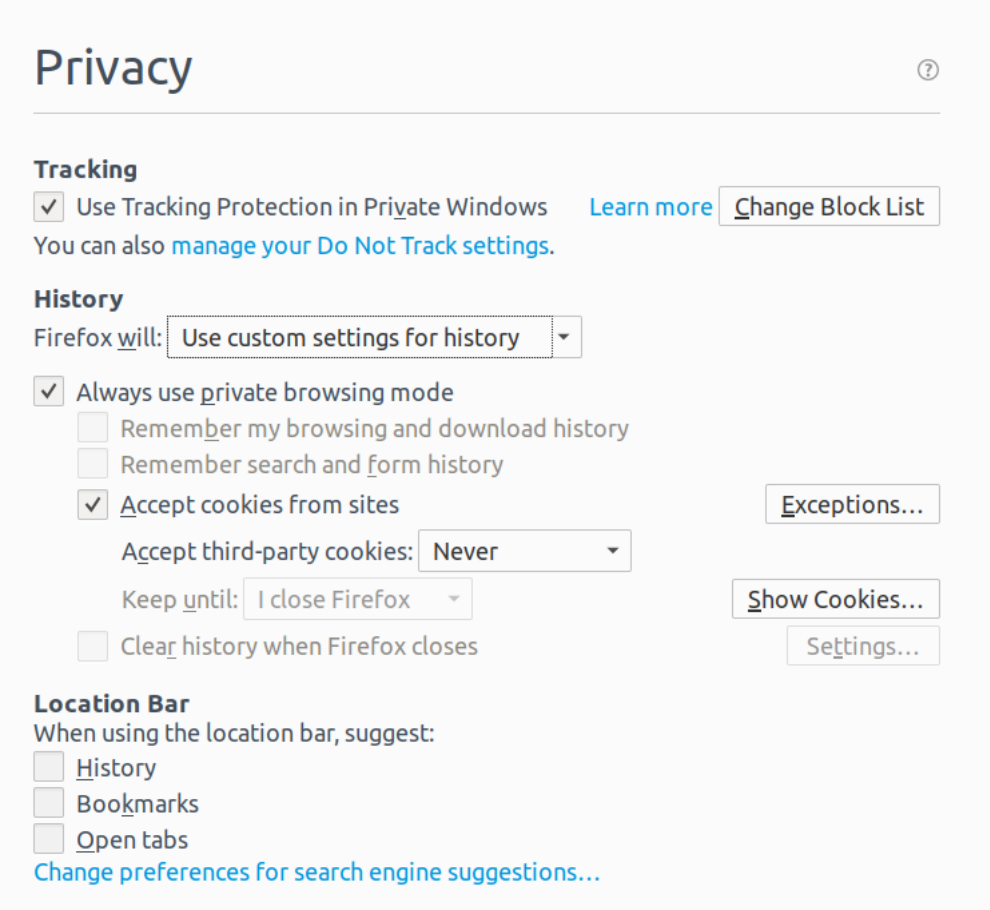

Figure A.1: The privacy settings offered in Firefox 49.0

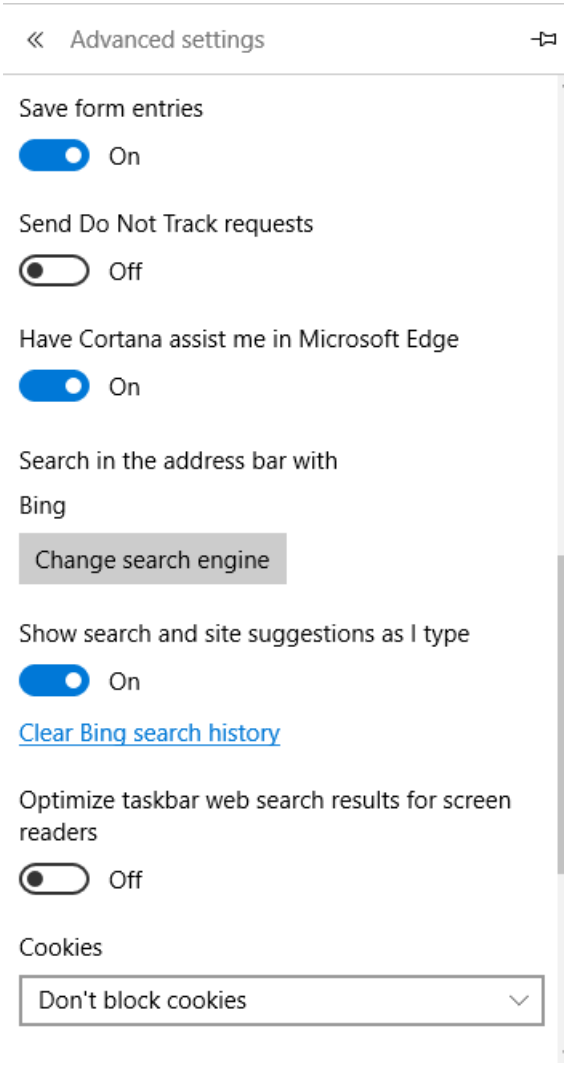

Figure A.2: The privacy settings offered in Microsoft Edge

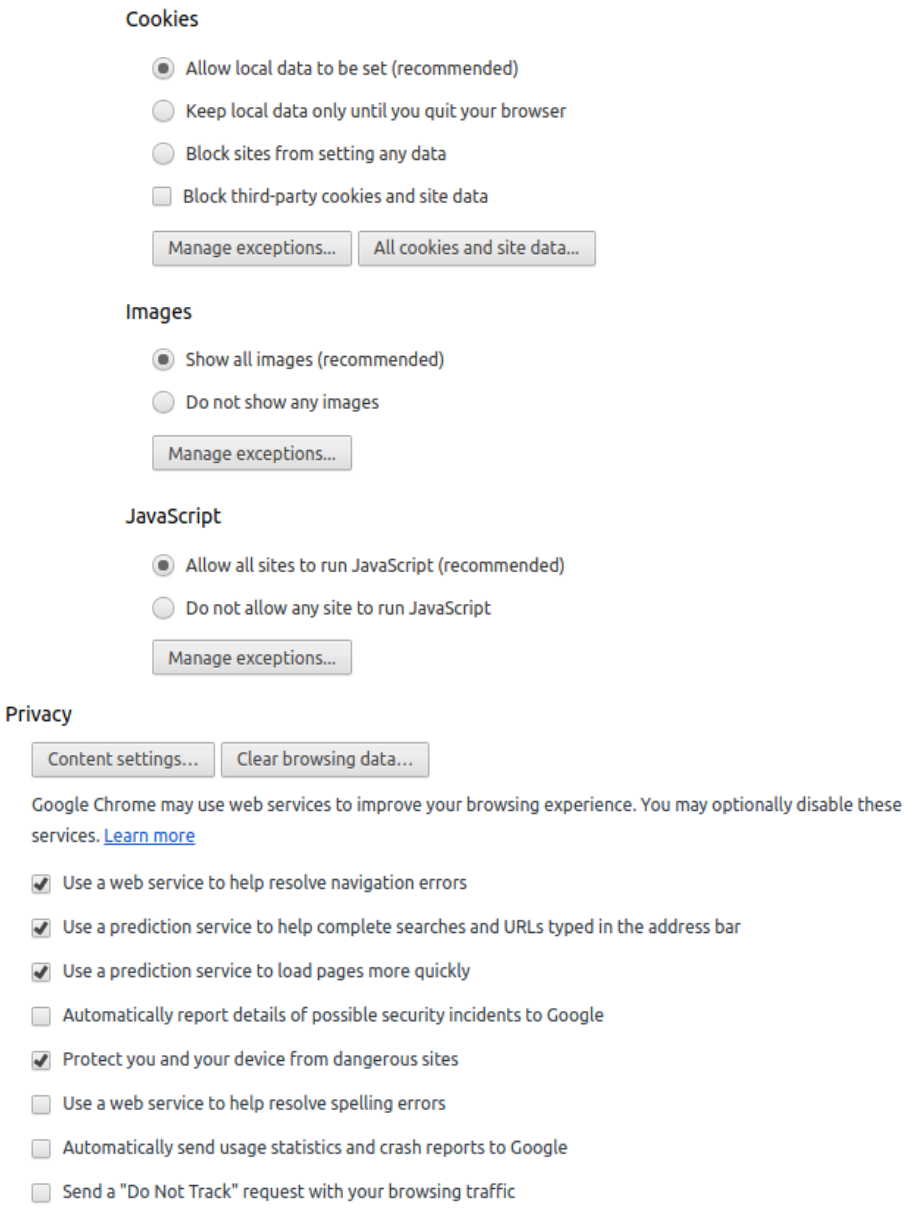

Figure A.3: The privacy settings offered in Google Chrome

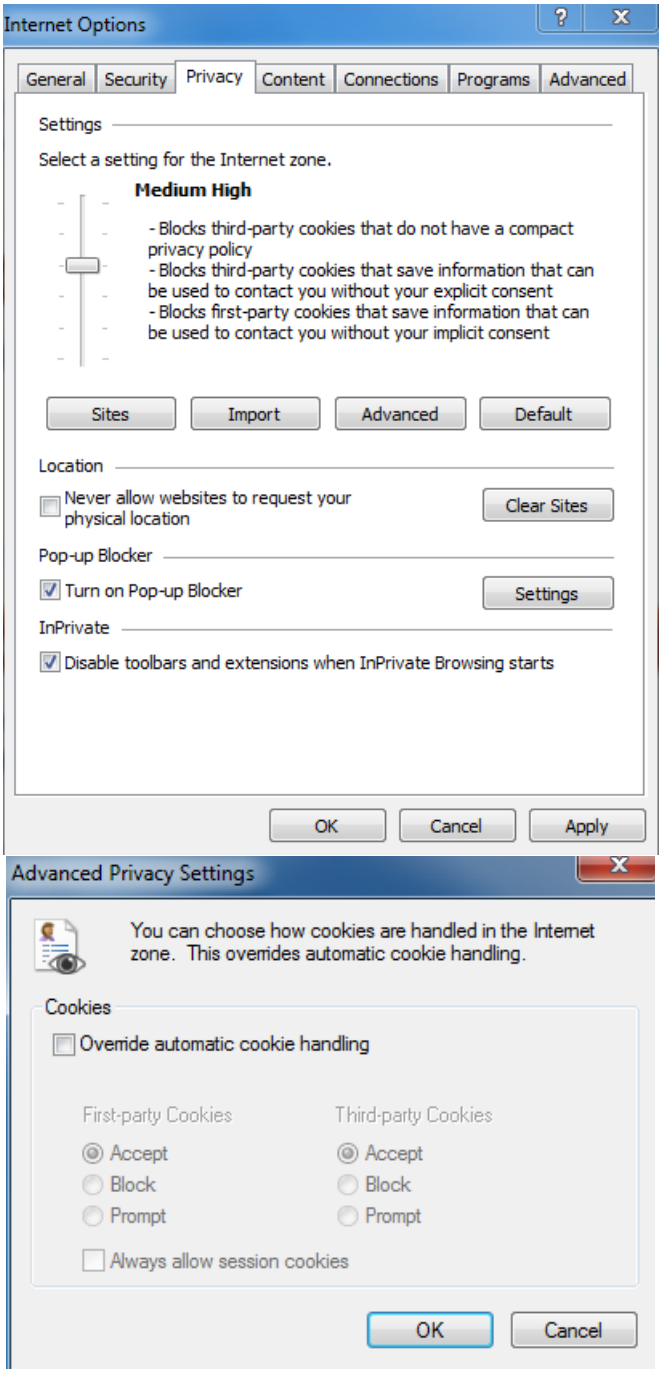

Figure A.4: The privacy settings offered in Microsoft Internet Explorer

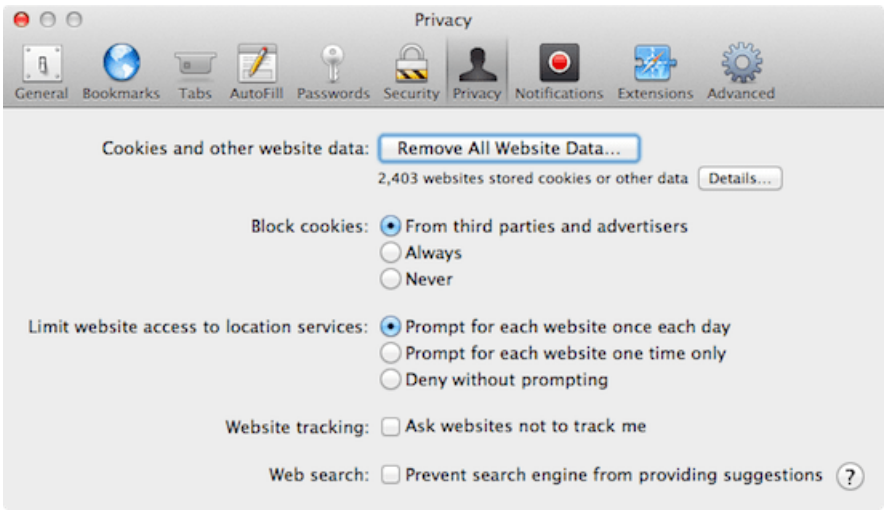

Figure A.5: The privacy settings offered in Apple Safari