# Has user privacy become a myth?

## HTML5 and the Privacy Act

Stephane Eyskens

December 8, 2014

The purpose of this article is to investigate whether the latest technological developments enter in contradiction with the current legislation about user privacy. It will try to understand to what extent the latest technologies, in particular HTML5, the ubiquity of browsers, a highly connected environment and the behaviour of users themselves could potentially lead to a world where privacy has become a myth. This paper will also shed some light on the root causes and the risks related to privacy leakage. At last, we will see how efficient, the existing precaution measures are in order to mitigate that phenomenon.

# Contents

# 1 Introduction

This paper will start by describing what is the concept of *Privacy* and what is *Personal Data* according to the Belgian legislation. We will then see why privacy related data is exploited by vendors and how the different commercial techniques and people behaviours evolved through time. The common thread of this article will be to put into perspective the current commercial practices on Internet and the concept of privacy as defined by the law in order to establish whether or not privacy violation is already effective today. At last, we will analyse how HTML5 features could be deviated in order to collect privacy related data at user's expense and what are the opportunities to mitigate that risk.

# 2 What is Privacy?

Personal data are all data that identify or can identify an individual directly, or at least that is how the Privacy Act defines them [pri [2014]]. The Privacy Act is the Act of 8 December 1992 on the protection of privacy in relation to the processing of personal data. This Act aims to protect individuals against abuse of their personal data. The rights and obligations of the individuals whose data are processed, as well as the rights and obligations of those processing the data have been established by the Privacy Act [pri [2014]].

*Processing of personal data* means any operation on the data such as collection, use, management or disclosure. The Privacy Act is a Belgian specific legislation and is made up of two Royal Decree (2001 & 2003) consisting of the transposition of European Directive 95/46/EC. The Decree of 2003 contains 74 articles describing how to handle privacy information. In a nutshell, these articles list a series of rules regarding *specified, explicit and legitimate purposes.*

Defining the notion of a *legitimate purpose* is rather difficult. Given the very formal descriptions above, it is important to distinguish two kind of disclosures, explicit disclosure is when a user explicitly discloses information such as uploading vacation pictures, expressing his mood onto a social media such as Facebook or Twitter. This information is explicitly set and shared by the user himself. Whether or not that type of disclosure goes against the Privacy Act is beyond the scope of this article.The unattended privacy disclosure however, is when a web site collects user privacy data at user's expense. We will focus particularly on this in the context of this article as it appears to be an unwanted *personal data processing* depicted earlier.

# 3 History of internet, user behaviour and commercial practices over the twenty past years

The purpose of this section is to describe how both the user behaviour and the commercial practices, advertising in particular, evolved over the past twenty years. It will also highlight the fact that user data represent the primary feed for most advertisers.

## 3.1 From the origins of Internet and its impact on user practices

Before Internet, end user computers were all disconnected desktops. Because of this, computers were isolated and the only way a piece of software could be present on a machine, was a duly setup performed by the computer owner. Privacy was therefore not as much at risk as today. Advertising was mainly done via traditional media such as television, radio and posters. The commercial purposes were probably the same as today but their penetration in people's life could not be done via computers.

Times have changed with the appearance of Internet. Netscape co-founder, Marc Andreessen, predicted that web browsers would one day render Windows obsolete [Wright [2009]]. Almost twenty years later, one can say that this statement is not completely true yet as many computers still run under Windows, but the evolution certainly tends to kill the desktop model in favour of a web model. This is even more obvious with the rise of Cloud Based services that appeared everywhere a few years ago, and that will probably remain for a long time.

Although browsers have not yet entirely replaced the good old desktop model, one can safely assume that most of today's applications are built upon web technologies, making browsers, the most frequently used client applications. The direct impact of this, is that most computers are directly or indirectly connected, giving much room to vendors to infiltrate in a way or another, end user's privacy for commercial purposes by leveraging more and more powerful browser features.

Since at least ten years, it must be noted that Internet is an integral part of people's life, as it appears that the average Internet user spends close to three hours per day online. This exceeds the time spent watching TV, which is about 1.7 hours per day [Nie et al. [2005]]. This was about Americans' use of Internet in 2004, before the era of Web 2.0 and Social Media such as Facebook, which have largely amplified this trend. Indeed, student's everyday life is deeply penetrated by Social Media[Debatin et al. [2009]]. Today, most people use Internet to do online banking, online shopping and a series of other activities such as communicating on Skype, interacting with friends on Facebook and tweeting on Twitter. Given that context, the risk of privacy leakage is largely increased, specifically because the users'information can be accessed, gathered, stored, data mined, linked, shared, contracted and potentially sold, for profitmaking purposes, and mainly, without permission or consent [Malandrino and Scarano [2013]].

The computers and mobile devices have become a new media to address consumers and this, mainly via their browser (and Apps). Nowadays, many web sites are financed by advertising. Their business model is sometimes entirely based on that. Therefore, being able to create reliable profile databases give trackers the opportunity to gather a significant amount of valuable information that can be sold afterwards to advertisers, thus increasing their revenues.

A very visible example of this, are the Facebook Ads, that anyone can buy to make advertising on Facebook. The role of the Facebook Ads is to build a very accurate targeted advertising [Cohen [2008]]. The price of an Ad depends on the volume of the targeted audience. The more people the Ad targets, the higher is the cost. Of course, all this happens at Facebook registered users expense even if they accepted the terms and conditions when registering to Facebook. Others than Facebook, also have their business model based on advertising. Sometimes, this can lead to unexpected and unwanted behaviours. For instance, press publishers may decide to privilege a topic over another based on the (larger) audience they can target in terms of advertising rather than on its real added value for the readers [Gabszewicz et al. [2001]].

The best evidence of privacy data exploitation is Google's answer to various accusations coming from other vendors about their approach to privacy. On a page [Goo [2012]], Google justifies itself on a form of a series of myths vs facts list. Another very symbolic statement is Eric Schmidt's, a long time Google CEO who declared : "If you're doing something that you don't want other people to know, maybe you shouldn't be doing it in the first place" [[Why, 2014]]. By adopting that kind of defensive behaviour in the one hand, and some kind of arrogance on the other hand, Google reveals its embarrassment when it comes to privacy.

Mark Zuckerberg, Facebook's current CEO, is certainly not outdone regarding inflammatory statements about privacy; he declared that "privacy is no longer a social norm" [[Why, 2014]]. These two declarations seem to indicate that big players such as Google and Facebook do really exploit privacy data, and although their policy is more or less explained [[Fac, 2012]], one can still be a little worried. How can we verify that Google and Facebook are totally transparent? Facebook exploits explicitly disclosed data but isn't Google part of the unattended privacy disclosure? What would happen to this data if these companies got hacked by malicious users?

## 3.2 Anonymous exposure does not prevent identification

According to the Privacy Act, data that cannot be related to an identified or identifiable person, is not considered as personal data. People might feel protected by the fact that they are not always visiting web sites in an authenticated manner, meaning by providing a user and a password. Even in the context of commercial web sites, which have a registration system enforcing user authentication, one could still feel more or less protected by using a nickname rather than his real name, but it is important to note that 87% of Americans can be uniquely identified from a birth date, zip and and gender information [Malandrino and Scarano [2013]]. There are numerous web sites prompting users to submit their birthdate, gender and/or zip code, and it seems that only a few information, is largely enough to formally identify someone. The Privacy Act is more lax regarding anonymity but, as we can see, anonymity is rather a fragile thing on the web. Should users always input dummy data when prompted to remain anonymous?

### 3.3 Reuse of privacy data to other ends than advertising

User privacy data is mainly used for advertising and commercial purposes. However, the data collected by vendors and advertisers could be reused to other ends. For instance, Facebook data has already been data mined by government agencies such as the police or the Central Intelligence Agency [Debatin et al. [2009]]. It also comes as no surprise that the NSA (National Security Agency) of the United States, is permanently collecting and monitoring information transiting via telecommunications, e-mails etc...and is allowed to do so since the Patriot Act.

In the light of what we have seen so far, there are some reasons to be worried regarding the acceleration of lost privacy. Today, people can still chose whether or not they want to be connected to Internet, in other words, they can still opt in or opt out. This is most probably due to the fact that the *old* generation has never been confronted to this connected world, implying automatically some kind of resistance, that is commonly accepted by the society for the time being...In a few generations, we can expect a change in the habits and customs and envision an impossibility to opt out.

Perhaps, actions such as paying taxes and performing administrative tasks will not be possible in a disconnected manner anymore. In the same vein, the latest evolutions of the Internet of Things (IoT), whose the purpose is to deliver more and more interconnected objects, that one use in our everyday life, will certainly leave fewer and fewer opportunities to opt out. Moreover, with the IoT, more sensitive data such as medical information, grooming rituals, viewing habits ...[Cre [2014]] will be gathered and expose individuals to important leakage.

If these questions of privacy are not fixed by then, it could become a real risk for people's freedom, in case our country (or even Europe) switches from a democracy to a dictatorship, or to an authoritarian regime. The technology could ease mass surveillance. These questions might sound alarmist but they deserved to be asked. The web might have become the modern form of archives, which could be exploited in a dangerous manner.

Back to our current reality, the main objective of data collection on Internet web sites, is to build user profiles in order to facilitate targeted advertising. But now comes this opened question : is *targeted advertising a legitimate purpose* or does it go against the

Privacy Act principles?

# 4 Does HTML5 threat the protection of personal data?

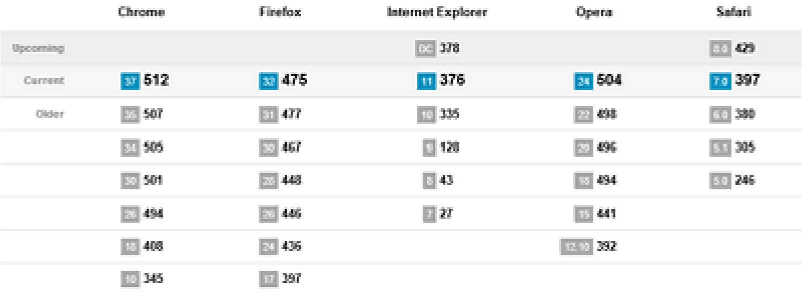
In the context of this article, we will only focus on a few HTML5 features that bring some additional APIs that may facilitate privacy leakage.

## 4.1 HTML5 origins

Mobile devices such as iPhone and iPad, clearly played a role in the elaboration and adoption of HTML5. The WHATWG (Web Hypertext Application Technology Group) was founded by individuals of Apple, the Mozilla Foundation and Opera Software in 2004 [wha [2014]]. While the WHATWG was working on a initial specification of HTML5, the W3C was going forward with XHTML2 but dropped it after a few years to focus on an aligned HTML5 effort with the WHATWG [Lubbers et al. [2011]]. The purpose of HTML5 is to build the next generation web applications.

## 4.2 HTML5, a built-in Privacy Trojan?

The *Candidate Recommendation* of HTML5 was released in 2012, however, this will still take time before browsers fully implement HTML5. Indeed, the *Proposed Recommendation* will only land in 2022 [Lubbers et al. [2011]]. Two years after the *Candidate Recommendation*, most modern browsers have already adopted HTML5 as illustrated by *figure 1*. The modern web applications can *already* leverage all the



| | Chrome | Firefox | Internet Explorer | Opera | Safari |
|---|---|---|---|---|---|
| Upcoming | | | DC 378 | | 8.0 429 |
| Current | 37 512 | 32 475 | 11 376 | 24 504 | 7.0 397 |
| Older | 35 507 | 31 477 | 10 335 | 22 498 | 6.0 380 |
| | 34 505 | 30 467 | 9 128 | 20 496 | 5.1 305 |
| | 30 501 | 28 448 | 8 43 | 18 494 | 5.0 246 |
| | 26 494 | 26 446 | 7 27 | 15 441 | |
| | 18 408 | 24 436 | | 12.10 392 | |
| | 10 345 | 17 397 | | | |

Figure 1: HTML5 Browser Support

features described later in this article.

### 4.2.1 Information Storage with WebStorage

Before diving into *Web Storage*, let's have a look at some techniques that are currently in use and that could still be used in addition to HTML5 in the future. Advertisers make use of Third-Party Cookies (TPC) to store small pieces of information in user's browsers. They are particularly used to track users among sessions and visits across different domains, and differ in that, from First-Party Cookies (FPC), which are stored in the same domain as the one of the visited page. According to a relatively old study, this is a very common practice, since 73% of the web sites place Third-Party Cookies in visitors browsers [Jamal et al., 2003]. The below graph, emited by Monica Chew, lead privacy engineer at Mozilla, tends to confirm the figures mentioned earlier.[Chew [2013]][1]

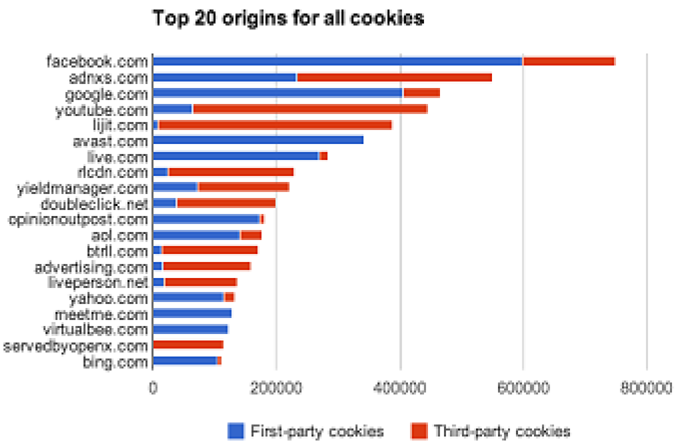


Figure 2: Top 20 origins for all cookies

Besides their number, these cookies have a quite long lifetime[2]. However, these cookies are limited in size, meaning that the amount of information they can contain is rather small.

[1] Monica Chew : "The graph shows the top 20 origins setting third-party cookies, responsible for 41.1% of third-party set-cookie events. adnxs.com belongs to AppNexus, an ad exchange. Facebook sets mostly first-party cookies, but because Facebook's social widgets are included on many sites, Facebook sets many third-party cookies (which may have originally been created in a first-party context). Of the top 20 origins, 18 primarily offer advertising services"

[2] Monica Chew : "The Set-Cookie HTTP header has an optional expiration time that tells the browser how long to keep the cookie. From the graph below, many cookies are long-lived, possibly longer-lived than the installation of the operating system or browser. 20% of third-party cookie expiration times were one week or less, and 51% of third-party cookie expiration times were longer than 6 months."

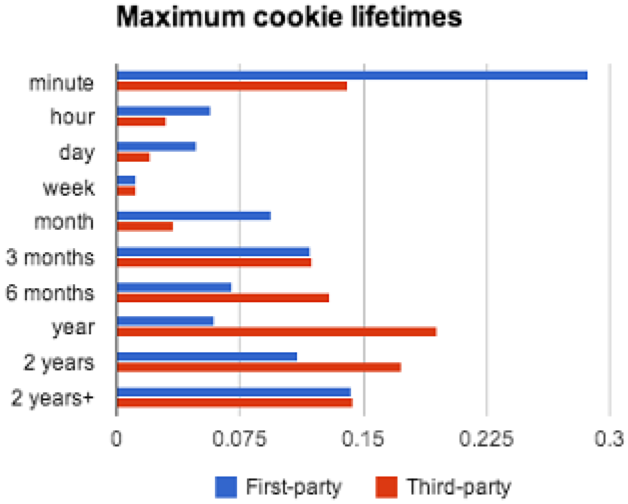


Figure 3: Maximum cookie lifetimes

When users took the habit of deleting their cookies, advertisers found a defence against it by leveraging Zombie Cookies [Pierson and Heyman [2011]], which are used as a fallback mechanism for TPC. In case TPC get deleted, web sites rely on the Zombie Cookies to recreate the TCP. Zombie Cookies are placed by plug-ins such as Adobe Flash and Microsoft Silverlight, which are allowed to write on a dedicated area of the user's file system, that is different from the cookie area. This is the reason why deleting cookies does not delete Zombie Cookies.

Nowadays, Adobe Flash and Microsot Silverlight are dead-end technologies, mainly because they are not supported on Apple devices such as iPhones and iPads. HTML5 has definitively superseded Flash and Silverlight, and comes with some *WebStorage* features that enable vendors to store a larger amount of information. One of this features is named the *sessionStorage*, which as its name indicates, allows web site owners to store session related data. Once the session expires, by closing the browser or after a variable amount of idle time, the data gets removed automatically. One could compare the *sessionStorage* to the FPC, with the difference that FPC can be stored beyond the duration of a session. Another difference, is that HTML5 storage features, do not send data back to the server, unlike cookies.

Therefore, *sessionStorage* as such, does not represent a real danger in terms of user profiling. However, *localStorage*, also part of the *Web Storage*, offers the possibility to store a larger volume of data, that varies according to the browser. As with Zombie Cookies, a third-party advertiser could use a unique identifier stored in its *localStorage* area to track a user across multiple sessions, building a profile of the user's interests to allow for highly targeted advertising [Hickson [2013]]. *localStorage* can be used as a backup of regular cookies, exactly as Zombie Cookies do.

For the time being, *localStorage* is not deletable directly from browsers like Chrome without specific extensions (or developer tools). So, why could HTML5's storage features be more dangerous than Zombie and TPC Cookies? Simply because they are already supported by all modern browsers, on all devices, including mobile ones. Moreover, Adobe Flash and Microsoft Silverlight were plug-ins that had to be explicitly installed by the user, while HTML5 is just plain part of the browser.

Initially, *localStorage* was not particularly designed to track user data, but rather to allow web applications to support an offline mode, by storing online services data. The potential problems related to this are going beyond basic user tracking. Indeed, users who consume online services, are usually well identified and have to provide a lot of details to the service provider. The sensitive nature of locally stored data, makes browsers a juicy target for cyber-attacks [Kimak et al. [2012]], but we will not develop further this aspect in the context of this article.

### 4.2.2 Tracking user's location with Geolocation

Before the release of the *Geolocation* feature, vendors were already able to locate a machine via its IP address, using some server-side code and this, in a silent manner. However, the *Geolocation* is much more accurate than the IP address technique as it can even detect the street you are in, especially when using a mobile device that has a built-in GPS [W3S [2014]]. Better than that, on GPS enabled devices, applications can leverage the *watchPosition* method to get the current position of a user as he is moving.

Until now, users are prompted whenever an application tries to determine their location as recommended by the W3C[Geo [2012]]. They are free to accept or decline the request. Depending on the nature of the application they are using, refusing to disclose their location, might cause the web site not to work properly. In other words, declining corresponds to taking the risk to work with a malfunctioning application.

This side effect might encourage users to always accept. Of course, letting the device disclose its location is equivalent to disclosing the user's location, thus privacy related information [Popescu [2013]]. Another interesting aspect of the *GeoLocation*, is that if a vendor tracks user location over time and saves this information in the *localStorage*, he will be able to establish a very accurate history of user movements; furthermore, he does no longer need a server-side database to store that information, because he can just rely on the user's device to do so. Geolocation adds another dimension to tracking, advertisers are not only able to determine user tastes and hobbies; they also know where you are and by crossing data, they might also know with whom you are.

### 4.2.3 Highly connected systems with the WebSocket

Before getting more details about the Websocket, it is important to step back for a while into the HTTP architecture. The HTTP protocol, uses a separate TCP connection whenever a browser gets a resource from a web server, which adds a significant overhead, especially in the number of network round trips involved in the operation [Padmanabhan and Mogul [1995]]. The purpose of the WebSocket specification, is to allow full-duplex communication channel between a web browser and a web server [Hämäläinen [2012]], using a single TCP connection.

The major difference with regular HTTP communication, resides in the fact that instead of sending multiple HTTP requests (thus multiple TCP connections) to get some data from a server, a browser can make a unique request and start an asynchronous silent communication with the server. Moreover, another major difference, is that a server can push some information to a browser, thus making a bi-directional communication possible. This feature is from far one of the most promising step ahead for real-time applications. So far, web browsers were using other techniques, such as *longPolling*, but that was not highly scalable.

What *webSockets* bring, is not only a highly performing communication, but also a very scalable one. How come could it affect privacy leakage? Technically speaking, in case of leakage, a significant bunch of information could be transmitted in a matter of milliseconds without the user even noticing. Functionally speaking, the *webSockets* represent the one step ahead that was still needed to completely mimic desktop applications, thus, delivering the final blow the desktop model.

By burying the desktop model, *webSockets* will contribute to the *WebOs*, thereby increasing the user dependency on the web.

## 4.3 WebOs and Cloud Services, leading to an increased web dependency?

Cloud computing is being the number one concept in IT industry, where all large IT companies such as Amazon, Microsoft, Redhat and Apple, are migrating their services [Abusaimeh]. More and more services are delivered via the Cloud, making the browser, the only required client. Today, most people have a Dropbox, Google Drive or a OneDrive for free. They also have their email provider in the Cloud. Google comes with plenty of Apps such as the Agenda, that makes the end user's life easy. Google Street View and Google Maps (also Bing Maps), are extensively consumed online services.

Beyond free user services, Cloud platforms such as Microsoft Azure and Amazon also come with very attractive functionalities and features targeting individuals as well as small and large enterprises. As an example, SharePoint online, that is part of the Office 365 bundle, offers not only online collaboration and document management services, but also close to infinite storage capabilities. Indeed, any enterprise can subscribe to an

Entreprise plan, whose the storage is about 500 Petabytes. In other words, if one take an average size of one megabyte per document, one could store up to one billion documents into a single Office 365 tenant [Off [2014]].

These figures demonstrate the power of the big players, who come with extremely competitive offers compared with what an enterprise can do with on-premises infrastructures. How is it related to privacy leakage? Well, this step ahead to a WebOs, makes users and enterprises more dependent to vendors. A new behaviour is slowly but surely entering in the habits and customs.

# 5 Opportunities for the protection of personal data

> The objective of this section is to analyse what are the precaution measures that can be undertaken in order to limit privacy leakage, from recommendations to behavioural habits.

## 5.1 Terms and Conditions

Terms and Conditions are a way to warn users about the potential use of their data. However, it is sometimes very tedious to read several pages at once and to understand all the statements. They have the virtue of informing users on how the data collected about them will be used (or not). While this is a useful piece of information, one must admit that users have only two choices: accept or decline but if they decline, they cannot use the application anymore, which is why they usually accept.

Terms and conditions are like license agreements, most of the people just accept them, often, without even reading them. It makes them quite ineffective to struggle against privacy leakage.

## 5.2 W3C Recommendations and initiatives

W3C makes a few interesting recommendations on how to limit privacy leakage. For example, they propose to bind *localStorage* to cookies, so that when end users clear their cookies, they also clear the *localStorage* in order to prevent sites from using the two features as redundant backup [Hickson [2013]]. This would eliminate the risk of using *localStorage* as a Zombie Cookies enabler. Another interesting recommendation is to build shared black lists, which would allow communities to act together to protect their privacy. It turns out that their own conclusion is not very reassuring with regards to the feasibility : "if a third party cooperates with multiple sites to obtain user information, a profile can still be created" [Hickson [2013]].

W3C also created the P3P protocol, whose the purpose is to enable web sites to express their privacy practices in a standard format that can be read by P3P-enabled user agents in order to inform users of site practices and to automate decision-making accordingly [P3P [2014]]. Unfortunately, the status of P3P is currently suspended due to a lack of support by browsers. Internet Explorer has a P3P Policy built-in support [IEP [2014]], but seems to be the only browser to support it among the leaders. W3C has not given up yet, providing there is a sufficient support for implementation in the future.

Could the lack of support and interest for P3P from vendors, be explained by the fact that they are themselves exploiting privacy data? They might not want to give users the appropriate user-friendly tools to struggle against privacy data collection? Besides the W3C, others also make suggestions on how to protect privacy. An idea is to encrypt the data that is stored in the *indexedDB*, which is where the *localStorage* data is located [Kimak et al. [2012]]. That would not prevent vendors from tracking users, but would offer a greater protection against malicious use of data.

## 5.3 The Web Origin Concept and the Same-origin Policy (SOP)

One of the major security concepts for browser-side programming, is the so called same-origin policy, which is defined in IETF RFC6454 [Hämäläinen]. This mechanism prevents scripts to access or send data from/to another domain. In the web world, a domain is made of :

```
[protocol]://[hostheader][:port]/
```

where protocol could be for instance *HTTP* or *HTTPS*, hostheader could be *hosta.silver-it.com* and port could be *80* which is the default port for HTTP communication while 443 is the default port of HTTPS communication. Browsers will prevent scripts from accessing data hosted behind a URL whose any of these three parts is different [Barth [2011]].

Although, in this example, the domain is *silverit.com*, if a script from the home page tries to access a resource that responds to **hostb**.*silver-it.com*, it will be rejected. However, over the past years, and due to the success of web applications, enterprises were in need of a way to workaround the same-origin policy, because it is very frequent that a page needs to communicate with a web service or a resource that finds itself on another domain. Therefore, some concepts such as CORS (Cross-Origin Resource Sharing), make it possible to communicate with another domain, providing the targeted web server allows this communication. Here again, advertisers (and attackers) could easily exploit this by sending any kind of data to their own web servers, for which CORS would be enabled beforehand; thus causing the browser to accept the cross-domain calls since it would receive an appropriate answer from the server.

The CORS mechanism can also be used with WebSockets [Hämäläinen]. Therefore, it is very easy to workaround the SOP and potentially send a ton of information silently to a remote domain. HTML5 also comes with the *PostMessage* API, that also helps to workaround the SOP. This API can be used to transmit messages between browser windows, even if they originate from different domains. This makes the SOP quite ineffective in modern applications.

### 5.4 Browser features and add-ons

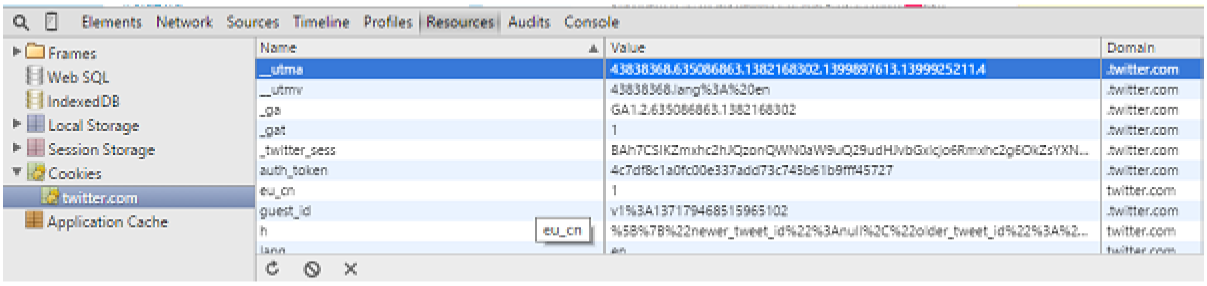
Technically speaking, it is possible to be very well protected against privacy leakage that would be caused by the HTML5 features described earlier. This is made possible by just by disabling JavaScript at the browser level. Indeed, these features can only be accessed by a client side scripting language, and the most commonly used is JavaScript. However, doing so would highly diminish the number of sites a browser would still be able to visit, since JavaScript is almost used everywhere. Opting for such an extreme solution, would most probably protect users against HTML5 features and scripting dangers, but cookies (FPC and TPC) could still be placed by servers. Indeed, FPC and TPC can be placed either by JavaScript (client-side), either by any kind of server-side technology. For instance, all the cookies that are flagged with the *HttpOnly* attribute, are server-side placed cookies. Such flagged cookies are not accessible from JavaScript. In most browsers, it is also possible to allow session-only cookies, meaning that these are destroyed after each visit. On top of blocking JavaScript, Google Chrome makes it possible to completely block TPC.

Internet Explorer comes with a three-zones (intranet/trusted sites/internet) security configuration, that enable users to specify different security settings per zone, allowing for instance to configure a flexible policy for Intranet sites and a more restrictive one for Internet sites. In Microsoft Silverlight and Adobe Flash plug-ins, users have the possibility to allow/refuse the use of locally shared objects. Besides the out of the box browser features, there are also numerous add-ons that can be used to block popups, cookies etc...but they usually come with a major drawback : they affect the overall performance and possibly the browser stability.

In summary, one can say that it is technically possible to restrict client-side and, to a lesser extent, server-side code to gather privacy related data. However, it turns out that this requires important skills, that most of lambda Internet users have probably not. Moreover, blocking browsers at their maximum often leads to unusable web applications.

## 6 Case-study : TPC and HTML5 storage in action

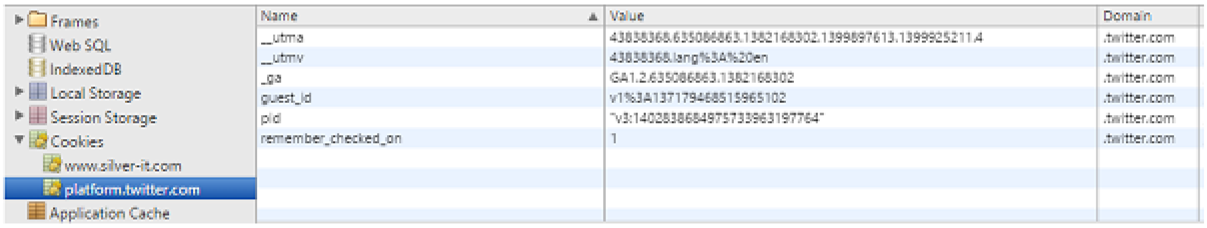
To clarify a little bit the points made earlier, here is a very concrete example of TPC in action. To illustrate this, we will use Chrome. By navigating to https://www.twitter.com/ and by hitting the F12 key, one can easily access the list of cookies present on the page, as illustrated by figure 4:



| Name | Value | Domain |
|---|---|---|
| __utma | 43838368.635086863.1382168302.1399897613.1399925211.4 | .twitter.com |
| __utmv | 43838368.lang%3A%20en | .twitter.com |
| _ga | GA1.2.635086863.1382168302 | .twitter.com |
| _gat | 1 | .twitter.com |
| _twitter_sess | BAh7CSIKZmxhc2hJQzonQWN0aW9uQ29udHJvbGxlcjo6Rmxhc2g6OkZsYXN... | .twitter.com |
| auth_token | 4c7df8c1a0fc00e337add73c745b61b9fff45727 | .twitter.com |
| eu_cn | 1 | twitter.com |
| guest_id | v1%3A137179468515965102 | .twitter.com |
| h | %5B%7B%22newer_tweet_id%22%3Anull%2C%22older_tweet_id%22%3A%2... | twitter.com |
| lang | en | twitter.com |

Figure 4: First-Party Cookie

On the left hand side, Chrome shows the *Cookies* section where only twitter.com is listed. Clicking on it reveals all the cookies that were requested/set by https://www.twitter.com. Looking closer, one can see a cookie named *_utma*. At this stage, _utma is a FPC and it's not possible to distinguish it from a TPC since Chrome lists it as a cookie belonging to the page's current domain (.twitter.com). Let's move on and visit http://www.silver-it.com. This time, we get two sub sections under Cookies :



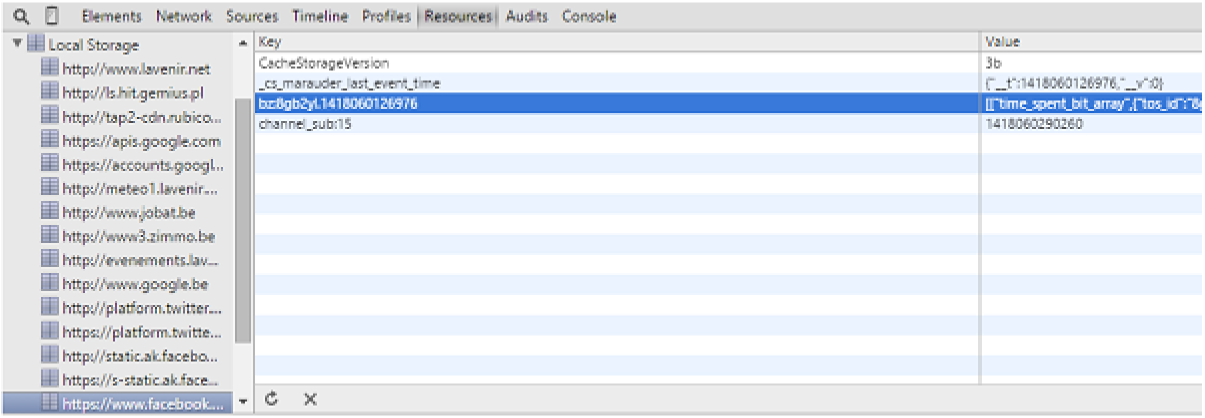
| Name | Value | Domain |
|---|---|---|
| __utma | 43838368.635086863.1382168302.1399897613.1399925211.4 | .twitter.com |
| __utmv | 43838368.lang%3A%20en | .twitter.com |
| _ga | GA1.2.635086863.1382168302 | .twitter.com |
| guest_id | v1%3A137179468515965102 | .twitter.com |
| pid | "v3:1402838684975733963197764" | .twitter.com |
| remember_checked_on | 1 | .twitter.com |

Figure 5: Third-Party Cookie

Cliking on *platform.twitter.com* reveals again a cookie named _utma although the requested page was *http://www.silver-it.com*. Moreover, _utma has the exact same value as previously. This value allows Twitter to recognize devices across pages and sessions. *_utma* is this time a TPC because it is set by Twitter although the page being browsed belongs to *silver-it.com*. How is that possible? There is no magic here, *http://www.silver-it.com*, as many web sites, makes use of the Twitter buttons which allow visitors to follow blog authors. Those buttons inject some Twitter related scripts that are most probably positioning these cookies. The exact same story goes for Google Analytics.

HTML5's localStorage and sessionStorage features can also be easily examined with

Google Chrome. When visiting *http://www.lavenir.net/*, and hitting the F12 key, one can see the content of the localStorage :

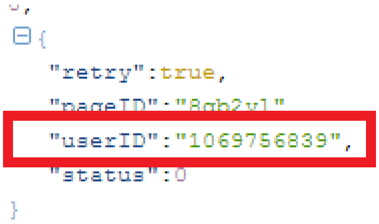


Local Storage
- http://www.lavenir.net
- http://ls.hit.gemius.pl
- http://tap2-cdn.rubico...
- https://apis.google.com
- https://accounts.googl...
- http://meteo1.lavenir...
- http://www.jobat.be
- http://www3.zimmo.be
- http://evenements.lav...
- http://www.google.be
- http://platform.twitter...
- https://platform.twitte...
- http://static.ak.facebo...
- https://s-static.ak.face...
- https://www.facebook...

| Key | Value |
|---|---|
| CacheStorageVersion | 3b |
| _cs_marauder_last_event_time | {"__t":1418060126976,"__v":0} |
| bz:8gb2yl.1418060126976 | [["time_spent_bit_array",{"tos_id":"8 |
| channel_sub:15 | 1418060290260 |

Figure 6: localStorage

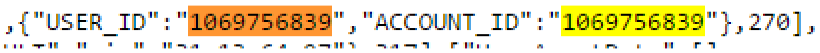
The highlighted key contains a stringified JSON value. Extracting it and pasting it to a JSON viewer, we can see many intresting things, among which, the fact that our Facebook user id is identified as shown by figure 7,



```
{
   "retry":true,
   "pageID":"8gb2yl",
   "userID":"1069756839",
   "status":0
}
```

Figure 7: Facebook data on Vers l'Avenir

although we didn't navigate to Facebook. Remember that we just visited *http://www.lavenir.net/*. Interestingly, when visiting *https://www.facebook.com* and when viewing the source of the page, one can find back the user id that was present in the localStorage area when visiting *Vers l'Avenir*. In the Facebook page, many references are found when looking for "1069756839" as for instance, this subset (for brevity) of a JSON object:



```
,{"USER_ID":"1069756839","ACCOUNT_ID":"1069756839"},270],
```

Figure 8: Facebook data on Facebook

This time, not only cookies were used but also HTML5's localStorage feature to store user related data across sites.

## 7 Conclusion

As explained in section two, the Belgian law specific to privacy, is the Privacy Act which is subject to interpretation, since most of its articles pertain to the notion of *Legitimate Purpose*, which is not explained. In section three, we demonstrate that users tend to use more and more Internet, which is now part of their life, thus, increasing their dependency to the connected world. We also demonstrate that the options to remain disconnected (opt out) are going to diminish in the future.

Later in section four, we saw that the browser is a modern trojan with regards to user privacy, because its native support of HTML5 makes it an ideal tool to collect and build user profiles in a silent manner, and because the browser is natively part of every mobile device. Its constant evolution, along with the growing commercial interests around the web, do not allow to envision a safe future with regards to user privacy.

In section five, we introduced the W3C, which is an independent organ led by Tim Berners-Lee, that proposes standards and recommendations. However, they are not entitled to dictate anything to vendors. A representative example, is the P3P protocol, which was not fully endorsed by vendors, and that is currently suspended. Besides W3C recommendations, browsers and add-ons can be configured to mitigate privacy leakage, but this requires users to have important skills and may lead to malfunctioning web sites.

Moreover, these restrictions do not apply to IoT so far, that is not only less restricted, but also less secure, thus exposing users to more dangers going beyond privacy exploitation as for instance, cybercriminals in 2013 who were able to remotely control a car's steering, brakes, acceleration, locks, and lights [Cre [2014]]. At last, because of the international nature of the web, it is very hard if not impossible to establish universal laws that would ensure a good usage of user privacy data in every single country of the world.

A legal arsenal is already in place today, but varies according to countries, and seems hard to enforce, especially because these technologies require important skills to verify whether they comply to existing laws. So, even if the laws were homogeneous and exhaustive, it would be still very hard to apply them from a technical perspective.

Many actions a browser can already do today, go against the Privacy Act, as one can safely assume that not all data collections are done for *legitimate purposes*. So, let us try to answer the initial question : Has user data become a myth? Our answer is : yes, it's already a myth today from a technical perspective and according to some studies mentioned earlier in this article. Could all this lead to excesses over the coming years?

Most probably, since it will become harder if not impossible for people, to opt out, thus causing all individuals to take part to this highly connected world and to use objects that know better their habits than individuals themselves.

## References

Commission for the protection of privacy, 2014. URL http://www.privacycommission.be/.

Alex Wright. Ready for a web os? *Communications of the ACM*, 52(12):16–17, 2009.

Norman H Nie, Alberto Simpser, Irene Stepanikova, and Lu Zheng. Ten years after the birth of the internet, how do americans use the internet in their daily lives. *Stanford Institute for the Quantitative Study of Society*, 2005.

Bernhard Debatin, Jennette P Lovejoy, Ann-Kathrin Horn, and Brittany N Hughes. Facebook and online privacy: Attitudes, behaviors, and unintended consequences. *Journal of Computer-Mediated Communication*, 15(1):83–108, 2009.

Delfina Malandrino and Vittorio Scarano. Privacy leakage on the web: Diffusion and countermeasures. *Computer Networks*, 57(14):2833–2855, 2013.

Nicole S Cohen. The valorization of surveillance: Towards a political economy of facebook. *Democratic Communiqué*, 22(1):5–22, 2008.

Jean J Gabszewicz, Dider Laussel, and Nathalie Sonnac. Press advertising and the ascent of the 'pensée unique'. *European Economic Review*, 45(4):641–651, 2001.

Busting myths about our approach to privacy, 2012. URL http://googlepublicpolicy.blogspot.be/2012/02/busting-myths-about-our-approach-to.html.

Why privacy matters?, 2014. URL http://www.ted.com/talks/glenn_greenwald_why_privacy_matters/transcript?language=en#t-176208.

Data use policy, 2012. URL https://www.facebook.com/policy.php.

The creepy new wave of the internet, 2014. URL http://www.nybooks.com/articles/archives/2014/nov/20/creepy-new-wave-internet/.

Whatwg faq, 2014. URL https://wiki.whatwg.org/wiki/FAQ.

Peter Lubbers, Brian Albers, Frank Salim, and Tony Pye. *Pro HTML5 programming.* Springer, 2011.

Karim Jamal, Michael Maier, and Shyam Sunder. Privacy in e-commerce: Development of reporting standards, disclosure, and assurance services in an unregulated market. *Journal of Accounting Research*, 41(2):285–309, 2003.

Monica Chew. Monica at mozilla in search of user sovereignty and tab sovereignty. 2013. URL http://monica-at-mozilla.blogspot.be/2013/10/cookie-counting.html.

Jo Pierson and Rob Heyman. Social media and cookies: challenges for online privacy. *info*, 13(6):30–42, 2011.

Ian Hickson. Web storage. W3C recommendation, W3C, July 2013. http://www.w3.org/TR/2013/REC-webstorage-20130730/.

Stefan Kimak, Jeremy Ellman, and Christopher Laing. An investigation into possible attacks on html5 indexeddb and their prevention. In *13th Annual Post-Graduate Symposium on The Convergence of Telecommunications, Networking and Broadcasting (PGNet 2012), Liverpool, UK*, 2012.

Html5 geolocation, 2014. URL `http://www.w3schools.com/html/html5_geolocation.asp`.

Geolocation api specification, w3c proposed recommendation, 2012. URL `http://www.w3.org/TR/2012/PR-geolocation-API-20120510/`.

Andrei Popescu. Geolocation API specification. W3C recommendation, W3C, October 2013. http://www.w3.org/TR/2013/REC-geolocation-API-20131024/.

Venkata N Padmanabhan and Jeffrey C Mogul. Improving http latency. *Computer Networks and ISDN Systems*, 28(1):25–35, 1995.

Harri Hämäläinen. Html5: Websockets. *Aalto University, Department of Media Technology*, 2012.

Hesham Abusaimeh. Cloud web-based operating system (cloud web os).

Sharepoint online: software boundaries and limits, 2014. URL `https://support.office.com/en-us/article/SharePoint-Online-software-boundaries-and-limits-8f34ff47-b749-408b-abc0-b605e1f6d498`.

P3p : The platform for privacy preferences, 2014. URL `http://www.w3.org/P3P//`.

How to deploy p3p privacy policies on your web site, 2014. URL `http://msdn.microsoft.com/en-us/library/ie/ms537341(v=vs.85).aspx`.

Harri Hämäläinen. How html5 affects the web privacy. *Aalto University*.

Adam Barth. The web origin concept. 2011.