Projet de Fin d'Études
2016 – 2017

[R|E]

Final Report

User discrimination in WEB

Student:

Leila Kuntar

Supervisor:

Nataliia Bielova

Abstract

User discrimination is a method of providing personalized content to the user. This content is founded on the user's profile. For instance, selling the same good at various prices optimized regarding the buyer's profiles is called price discrimination. The method has recently been used by some e-commerce web sites, for example, hotel booking or general retailers. A higher price will be usually suggested to a buyer with higher revenues. For example, they deduce from a brand of the computer used by this buyer. Search results will normally suggest pricier products to a buyer with higher interests. It is deduced from the buyer's previous browsing history. A targeted advertisement is another example where user obtains personalized ads. These ads are founded on the user's interests and profile. Researchers have found that females receive fewer ads with high paying jobs lately. As a general rule, 'user discrimination' is any method employing a user's profile to supply tailored content that varies from price and search discrimination to targeted advertisement. This kind of discrimination can be acceptable if it is obvious to the users, but it may be a problem if it is hidden from the users. Collection and sharing of users' personal information are requested that infringes privacy if there is no user agreement.

TABLE OF CONTENTS

INTRODUCTION

FRAMEWORK/CONTEXT

Tracking pixels (known as «Web beacon» or 1×1 pixel image) are incredibly useful for tracking stats of user’s behavior on the Web. These stats are not usually trackable. It would seem harmless or invisible for user who may be surprised that it is one of the ways to monitor his online behavior. Beacon images allow to track users reading email with images turned on, advertising impressions and checkout pages where JavaScript and POST requests are denied, but where it is possible to embed an image.

Web tracking with beacon images mostly works like this:

- An image resource is embedded in page or email.
- The source attribute of the picture points to a URL on tracking server.
- The client (browser, email application) then asks for the URL in the usual way.
- The tracking server first goes through a set of instructions noting down the HTTP request's details and collecting these stats in a database.
- The server then returns an invisible picture – commonly a 1×1 transparent pixel.

The question arises as to whether or not web tracking with beacon images would work without cookies.

MOTIVATIONS

Privacy has recently become a striking issue in the context of electronic commerce websites that track consumers visiting them. This type of tracking may occasionally be useful to the consumer, for instance, when an e-commerce site offers a shopping cart for customer purchases. However, collecting user's data regarding his entire visit of a site seems harmless unless the user surfing the Web at about 2 a.m. receives advertisements for insomnia cures, signifying the presence of privacy vulnerability.

Web tracking is getting more attention, it is known that “several trackers can each capture more that 20% of a user’s browsing behavior” [1]. It has been found that Doubleclick can track a user across ~39% of the pages he visited. As many users are logged into Google and Facebook accounts, this tracking doesn’t seem anonymous. In fact, the user doesn’t realize what kind of information about his activities is being provided to third-party organizations.

Also he has pure knowledge of the purposes of collecting information: selling aggregated information to third parties, targeting advertisement, malware distribution and creation of user social graph. The user has no control mechanisms of manipulation of his personal information such as navigation and search history within different sites. Nowadays consumers seem to be lost: current control mechanisms and countermeasures are not always efficient and are hard to use.

CHALLENGES

A web page may contain huge numbers of pictures. It is necessary to develop the automated way of revealing whether or not the picture is a beacon.

Web tracking can be done in different ways and use various features of web pages/browser. Any resources that allow content downloading from other domains can be used for tracking:

- Images, scripts styles;
- Pages within iframes;
- Flash objects.

All these elements can be added dynamically to a web page with no restrictions. The collected data can be serialized and sent back to a server at GET request or a URL path.

Another channel of distributing personal data is http headers. Trackers can use HTML redirects and set cookies via HTTP headers. The following elements can be serialized in http headers:

- User referrer (a web page that the user visits);
- User agent.

Moreover, user IP address can be resolved by means of http connection on the server. A web request must be blocked before a connection is created for protection reasons.

GOALS

The goal of this project is to study whether beacon images are the most prevalent form of tracking today. The beacon monitoring system must be able to provide a reliable method of judgment as to whether or not a particular image is employed to track the user.

It is essential to find out what companies provide web beacon tracking and what standard way to track with beacons is. Further, methods currently used for tracking will be analyzed to ensure that existing solutions for defense against web tracking are effective.

STATE OF THE ART

In this section, we will explore existing methods for preventing web tracking.

THIRD PARTY COOKIE BLOCKING

Blocking third-party cookies are usually recommended as a first line of defense against third-party tracking. But this defense is inefficient for some reasons. First different browsers can implement third-party cookies blocking by means of various policy rules. All well-known browsers implement third-party cookies blocking. But it is not obvious because a browser may block the setting of third-party cookies and their sending at the same time. In that case, for example, MySpace can set a first-party cookie when user visits myspace.com. This cookie, once set, is available to MySpace from a thirdparty position when script is embedded on another page. Turning off the browser's cookies will prevent Web beacons from tracking the user's activity. But the Web beacon still will account an anonymous visit, though the user's unique information will not be recorded.

DO NOT TRACK

The Do Not Track (DNT) header is the proposed HTTP header field that requests that a web application disables either its tracking or cross-site user tracking (the ambiguity remains unresolved) of an individual user. It would seem that this is a standardized way for users to opt out of web tracking by appending a DNT=1 header to outgoing requests. But there are no legal or technological requirements for its use when it is enabled by default on browsers. Do Not Track is barely a policy technique that requires tracker acquiescence, providing no technical backing or enforcement. As such, websites and advertisers may either honor a request or completely ignore it in cases where it is automatically or manually set. On the other hand, there has been some concern [2] that pervasive opt-out of tracking will create a divided web, in which visitors who opt out of tracking will not be provided with the same content as other visitors.

CLEARING CLIENT-SIDE STATE

One conceivable solution to avoid tracking is to continually clear the browser's client-side state, recurrently receiving new identifiers from trackers [3]. This may be a sufficient solution for some trackers, but it cannot protect users against the cross-site tracker visited directly by the user in other cases. Also known as "evercookies", respawning technologies can actively besiege user's deliberate attempts to start with a clear profile by abusing different browser storage mechanisms to restore removed cookies [4]. Many trackers set unique identifiers in HTML5 LocalStorage and duplicate these values in cookies, or in Flash LSOs, even in exotic ways like cache Etags. As the same identifier is stored in multiple locations, the possibility of respawning is incredibly raised. Respawning it not an unusual occurrence in modern realities and has been observed several times in the real user's experience [5]. Besides it was shown [6] that "fingerprinting techniques can reidentify a large fraction of hosts with fresh cookies".

BLOCKING POP-UPS

Most browsers today block popups by default. However, websites can still open popups in response to user's clicks. There are other methods that can be used to force a user to visit directly a website. For example, they redirect the user's browser to the tracker's domain and back using javascript. Moreover, on account of business relationships between the tracker and the embedding site, it is possible to redirect immediately to a full-page intermediate advertise controlled by the tracker. These behaviors are hard or impossible to block as they are used throughout the web for other legitimate purposes.

MISCELLANEOUS

Besides the standard ways of protection considered above, there are also other more or less extensive techniques. It is simply to use private browsing mode, although it does not primarily address the threat model of web tracking [7]. Private browsing mode does not aim to keep a user's browsing history private from remote servers.

Another possible way to prevent web tracking is installing browser extensions. Nowadays free and proprietary software is being actively developed. It represents extensions for browsers, which try to prevent web tracking by analyzing server requests and blocking undesirable content. For example, PixelBlock will show a red eye warning when an email is using a beacon image to determine if a receiver checks his mail. It is also possible to change mail settings in order to disallow email messages to download images.

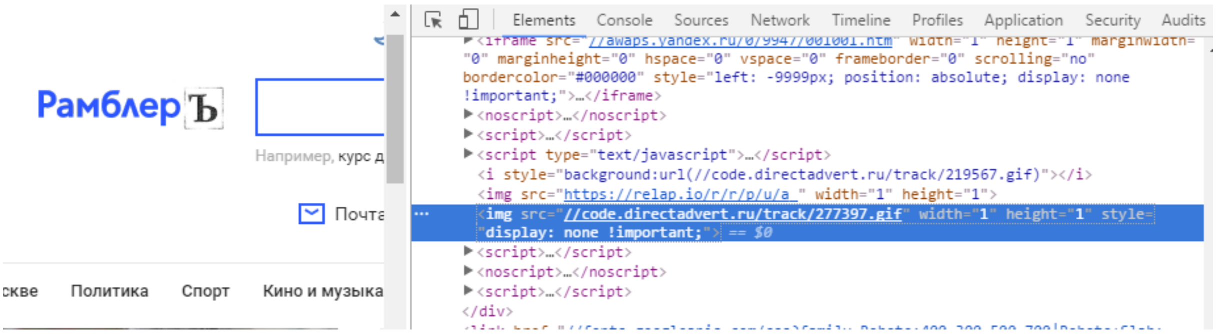
Disabling JavaScript can be effective against tracking behaviors that require API access to leak cookies, but it is the bluntest defense against trackers that can use HTML redirects and set cookies via HTTP headers. Besides disabling JavaScript, there is a significant effect on the browsing experience that becomes an unworkable option for most users. By the same token, some trackers simply use tags to fetch beacon images, even if more complex scripting techniques are not available.

APPROACH

GENERAL

In this work, I analyze a collected data from the top 800 websites, to determine if the picture is a beacon; to learn more about websites using beacon images and for what purposes; to understand whether or not beacon images are the prevalent form of tracking nowadays, what standard way to track with beacons is and if they rely on JavaScript or cookies.

It is considered that a Web beacon has a width of and a height of 1. Figure 1 shows the example of such tracking image. Though this kind of images may have non-standard sizes as we will see later, the main condition for the user is invisibility.



```
▸<iframe src="//awaps.yandex.ru/0/9947/001001.htm" width="1" height="1" marginwidth=
"0" marginheight="0" hspace="0" vspace="0" frameborder="0" scrolling="no"
bordercolor="#000000" style="left: -9999px; position: absolute; display: none
!important;">…</iframe>
▶<noscript>…</noscript>
▶<script>…</script>
▶<script type="text/javascript">…</script>
 <i style="background:url(//code.directadvert.ru/track/219567.gif)"></i>
 <img src="https://relap.io/r/r/p/u/a " width="1" height="1">
 <img src="//code.directadvert.ru/track/277397.gif" width="1" height="1" style=
 "display: none !important;"> == $0
▶<script>…</script>
▶<noscript>…</noscript>
▶<script>…</script>
</div>
```

FIGURE 1 EXAMPLE OF A BEACON IMAGE IN THE SOURCE CODE OF WEB PAGE

In this work it has been analyzed a collected data from several JSON files gathered by a web crawler. These files contain all links to the first and third-party images from top websites. Information about a website's rank - that is calculated using a combination of average daily visitors and page views over the past month - can be found on ALEXA website[1].

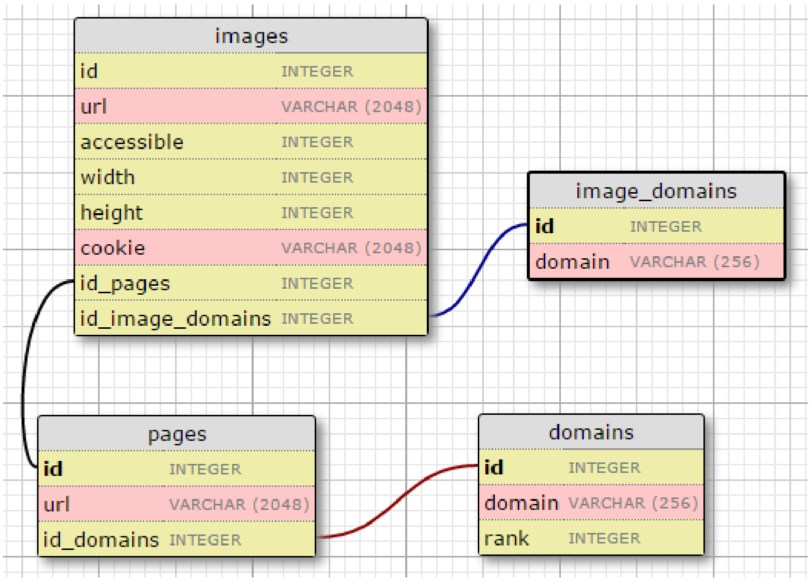
A script has been developed as a means to parse JSON files and to retrieve a third-party image URLs. This type of images belongs to the third-party tracker, i.e. its tracking code is included or embedded in other site.

Developed script parses the JSON file in order to find a set of images links that are only used to monitor user's activity from a third-party position. To determine what image is a third-party, a top level domain name is being extracted from image URL with a help of Python

[1] http://www.alexa.com/topsites

library tld[2]. It may rarely happen that domain names are not recognized or recognition is not precise. It is a restriction of library and is out of this work.

The obtained data about beacon images is stored in a database SQLite, the schema of which is shown in Figure 2.



images	
id	INTEGER
url	VARCHAR (2048)
accessible	INTEGER
width	INTEGER
height	INTEGER
cookie	VARCHAR (2048)
id_pages	INTEGER
id_image_domains	INTEGER

image_domains	
id	INTEGER
domain	VARCHAR (256)

pages	
id	INTEGER
url	VARCHAR (2048)
id_domains	INTEGER

domains	
id	INTEGER
domain	VARCHAR (256)
rank	INTEGER

FIGURE 2 DATABASE SCHEME

In table “domains” script stores a domain name of website and its rank; in table “pages” – all pages that belong to this domain; in table “image_domains” – top level domain name of image’s URL. Information about a third-party image is stored in the table “image” containing the following fields:

- id – a primary key of table images;
- url – a link to an image;
- accessible – a logical value (1 – url is accessible, 0 – otherwise);
- width – a width of image in pixels;
- height – a height of image in pixels;
- cookie – a cookie values that are set with URL response;
- id_pages – a foreign key to the table pages;
- id_image_domains – a foreign key to the table image_domains.

The script gathers images in simulated conditions as if all cookies have been cleared. Therefore, there were no any cookies on request to servers during experiments. And a

[2] https://pypi.python.org/pypi/tld/0.7.8

database field with cookie from server response is not as useful as suggested during database scheme development stage. However, it can be helpful because there are links that lead to corrupted images. Thus, it is impossible to distinguish a size. In this case, if server returns a corrupted image with setting cookies it will seem to be a beacon tracking.

For statistics, it is interesting to collect accessible field to examine the degree of stability of links to beacons. Different types of errors may occur when downloading images:

- Max retries exceeded;
- 404 (not found);
- Read timed out;
- Exceeded 30 redirects.

We consider that beacon image is a picture that has a width of 1 and a height of 1. As we will see further on, it is useful to consider that image with a width of 0 and a height of 0 is also a beacon. The script analyzes images with the method “Check the size” (also called as method №1) and puts collected information in database №1.

The script also collects information in another similar database №2 having only one difference: instead of the fields “width” and “height” it has a field “content length". For this database, the script analyzes images in a different way. It will check the content type and the content length of HTTP response. The method “Check the content length” (we call it also as method №2) shows that if there are a content type equal to “IMG” and the content length less or equal to the parameter x this image will be a beacon.

PARAMETER CHOICE

According to the method “Check the content length”, we consider that image is a beacon if the content length of response is more or equal to 0 and less or equal to x.

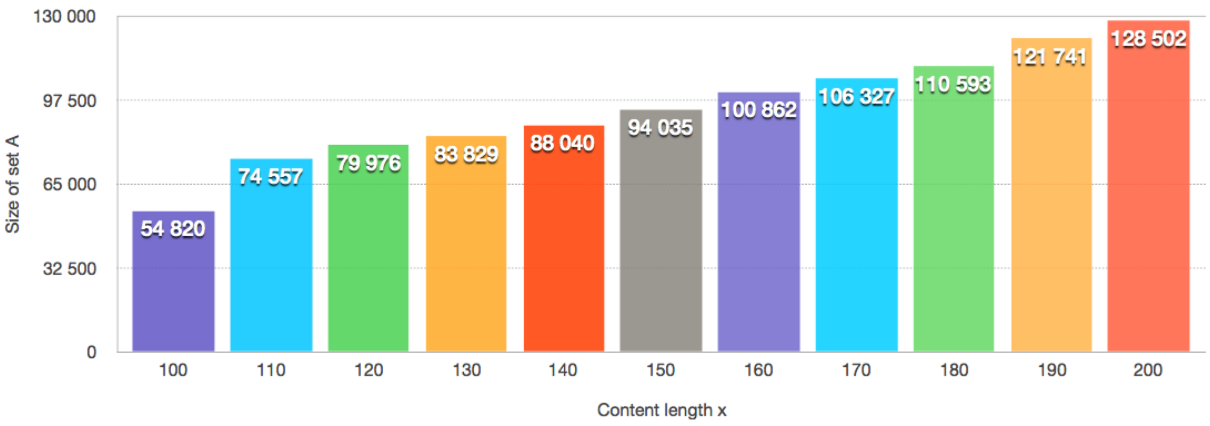


FIGURE 3 HOW MANY BEACONS WERE FOUND BY METHOD №2 BUT NOT BY METHOD №1?

In Figure 3, we can see how many beacons have been found by the method “Check the content length” and not by the method “Check the size” (the set A) depending on the maximum content length x.

Thus, with an increase of the parameter x, the size of set A also goes up. However, as the method “Check the content length” is a heuristic method, there are false positives results. To compare the two methods, we have chosen the minimum content length of 100. In this case, a difference between the two sets of beacons has a minimum value.

METHODS COMPARISON

As shown in Table 1, the statistics on the two databases are not the same.

	Method “Check the size”	Method “Check the content length”
Count of beacons	2 431 277	2 316 179
Count of 3rd party images	5 873 372	5 830 542

TABLE 1 STATISTICS FROM THE DATABASES

It is because the script puts image information into database only if the conditions are met. In case of the method №1, a downloaded image should be a third-party image with any type (PNG, GIF, JPEG, SVG, etc). In case of the method №2, the response with the third-party image should have the fields “content length” and “content type” in HTTP header. But as the “content-length” and “content-type” are not the necessary fields in HTTP header, a number of images in the two databases is different.

To find the symmetric difference between the two methods, the sets of 1 000 000 beacons have been compared by the developed script. This script looks for such beacons that are presented in one database but are absent in the second database. As operations of execution SQL requests take a lot of time, a selection has been restricted to one million. Besides, it is enough without checking all sets of beacons.

The size of relative complement of the method №1 in the method №2 is 40 647 (Figure 4).

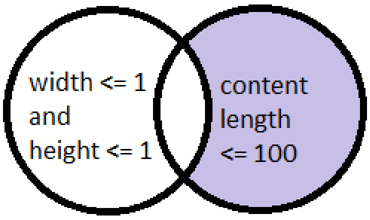


FIGURE 4 THE RELATIVE COMPLEMENT OF METHOD №1 IN METHOD №2

The size of relative complement of the method №2 in the method №1 is 101 731 (Figure 5).

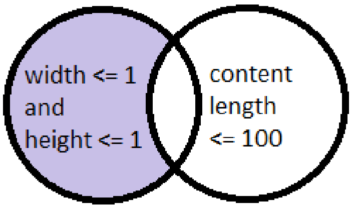


FIGURE 5 THE RELATIVE COMPLEMENT OF METHOD №2 IN METHOD №1

Therefore, the symmetric difference of the two sets with beacons contains 142 378 beacons. In order to understand what method is more appropriate for the automated definition whether or not the image is a beacon, it is necessary to manually check several links. In Table 2, we can see summary information about beacons that have been detected by the method №2.

Page	URL or part of URL	Content	Width	Height	False Positive
http://www.cdiscount.com/auto/coffres-de-toit/l-13359.html	http://i4.cdscdn.com/RWD/header/nempty.gif	43	4	1	Yes
http://stanford.edu/search	http://www.google.com/uds/css/clear.gif	58	9	9	Yes
http://www.gmanetwork.com/radio	https://tpc.googlesyndication.com/pagead/000000_new_ico.gif	74	13	13	Yes

http://nypost.com/2016/11/27/fidels-mistress-was-recruited-to-kill-him-he-seduced-her-instead/	*http://pixel.wp.com/g.gif?blog=56757169&v=wpcom&tz=5&user_id=0&post=10564540&subd=thenypost&host=nypost.com&ref=&rand=0.16372917289845645*	*50*	*6*	*5*	*No*
https://www.engadget.com/2016/11/09/planet-earth-might-be-the-biggest-loser-under-president-trump/	*https://s.sa.aol.com/b/ss/aolwbengadget,aolsvc/1/JS1.4.3/s72453359663486?AQB=1&ndh=1&pf=1&t=28%2F10%2F2016%201 [....]*	*43*	*2*	*2*	*No*
Page	**URL or part of URL**	**Content Length**	**Width**	**Height**	**False Positive**
http://time.com/4579790/jimmy-kimmel-kids-politically-correct-thanksgiving/?xid=homepage	*https://beacon.krxd.net/usermatch.gif?kuid_status=new&partner=google*	0	-	-	No
http://ck101.com/space-username-bpd.html	*http://sla.ckcdn.com/staticx3/image/common/px.png*	69	50	1	Yes
https://tabelog.com/en/hyogo/A2801/A280101/28041271/	*https://kakakucom.112.207.net/b/ss/kakakucomtabelogcom/1/H.24.1/s77363805295899?[...]*	*43*	2	2	No

TABLE 2 WHAT WAS FOUND BY METHOD №2 BUT NOT BY METHOD №1?

As we can see, there are beacons that have the content length less than 100 bytes but dimensions more than 1px. Also, there are false positives, for example a response has the content length less or equal to 100, but a delivered image is visible and does not seem to be a beacon image. In addition, we do not consider images from content providers used to collect statistics within the 1st party site and not from the 3rd party position.

Page	**URL or part of URL**	**Content length**	**Width**	**Height**	**False Positive**
http://www.prothom-alo.com/technology-	*https://cm.g.doubleclick.net/pixel?google_nid=mediamath&google_cm&google_hm=*	*170*	*1*	*1*	*No*

research	*w2jypcrbri76brk8qkqyw*				
http://news.ameba.jp/hl/20161128-587/?adxarea=endBb	*http://sync.fout.jp/sync?xid=appvador&uid=efb1bbc1-a6bd-4db7-b3af6350a3d0deco*	-	*1*	*1*	*No*
http://www.gazetaexpress.com/	*http://cookie.veinteractive.com/pixel.png*	*2792*	*1*	*1*	*No*
http://softbobo.pixnet.net/blog/post/12765887	https://pixel.everesttech.net/1x1	128	1	1	No
http://www.banggood.com/Wholesale-Stuffed-Cartoon-Toys-c-3304.html	*http://sync.madnet.ru/google/sync/*	*180*	*1*	*1*	*No*
Page	**URL or part of URL**	**Content length**	**Width**	**Height**	**False Positive**
http://www.elmundo.es/lo-mas/noticias-mas-leidas.html	*https://tpc.googlesyndication.com/simgad/16905052285678720172*	*807*	*1*	*1*	*No*
https://www.amazon.ca/b/ref=nav_shopall_gno_toys_hobby?ie=UTF8&node=6303777011	*https://images-na.ssl-images-amazon.com/images/G/15/acs/ux/transparent.png*	*951*	*1*	*1*	*Yes*
https://my-hit.org/film/a9/	*https://sync.pool.datamind.ru/image?source=adwise&id=*	*181*	*1*	*1*	*No*

TABLE 3 WHAT WAS FOUND BY METHOD №1 BUT NOT BY METHOD №2?

Summary information about beacons that have been detected by method №1 is shown in Table 3. As we can see, there is only one false positive being a consequence of restriction for the detection of 3rd party images domains. The script does not recognize that domains *images-na.ssl-images-amazon.com* and *amazon.ca* are related. Though the image is a beacon, we do not consider it because it is used for statistics collecting only from the 1st party position.

Therefore not all beacons have a width of 1 and a height of 1 and not all servers' responses with beacons have content length less or equal to 100 bytes. There are images that have dimension equal to 1px but content length more than 100.

There are situations when method that checks the content length works fine while method that checks the image size doesn't work. That happens because there is no image in server response. HTTP response 204 means that there is no content, but cookies are being sent through image request that means that it is similar to web tracking via beacons.

Thus, for better recognition of beacons images it is possible to improve the method “Check the image size” by checking content type and status of server response. In this case, we consider that image is also a beacon if the content type in HTTP header contains the substring “*image/”* and if a server returns status 204.

By using the method “Check the content length”, precision of beacons detection decreases because not all responses have the fields “content length” and “content type” and they can keep wrong information. Moreover, several beacons with dimension 1px are being sent in response with enormous content length because a type of image is PNG having a huge metadata. So to improve the method №2 it is possible to dynamically vary the parameter *x* depending on a type of image.

RESULTS

In this section, we analyze data from database in order to answer the questions how many popular websites use beacon images for web tracking and whether beacon images are the most prevalent form of tracking today.

GENERAL STATISTICS

For the sake of precise detection of beacons the improved method “Check the image size” is used. Thus it is considered that a beacon – is such 3rd party image that has dimension 1px or 0px. As it is described above in case if status of server response is 204 and if content type in HTTP header contains the substring “*image/”* the script considers that image is also a tracking pixel and puts in database the values: width = 0 and height = 0. Though the number of such images is 37 294 that is 1.53% of beacons.

Count of links to images that have been checked	**8 586 314**
Count of 3rd party images	5 873 372
Count of beacons	2 431 277
Count of errors	16 385

TABLE 4 STATISTICS

Interestingly, the amount of discovered beacons is 41% of total amount of 3rd party images in database (Table 4). The link to images that returns some error (e.g. 404) is also stored in database with the field: accessible = 0.

Count of domains	800
Count of domains with at least one beacon	760
Count of pages	124 214
Count of pages with at least one beacon	111 442
Count of image domains	4 348
Count of image domains where image is a beacon	1 325

TABLE 5 DATABASE STATISTICS

As we can see in Table 5 a huge amount of pages (89.7%) contains beacon images. And 40.37% of image domains are used for providing beacons. Even more remarkable, in 5% of domains beacon images have not been found. It may mean that these domains use other ways of web tracking (e.g. fingerprints) or that the provided method does not distinguish beacons that they use (e.g. beacons with not standard dimension 2px).

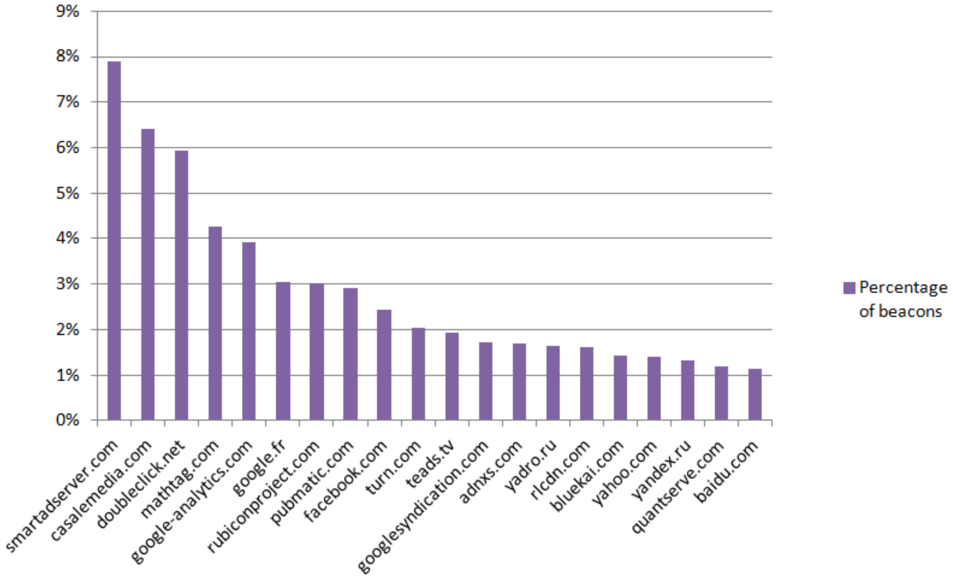


FIGURE 6 IMAGE DOMAINS – PERCENTAGE OF BEACONS

In Figure 6, we can see how many beacons out of 2 431 277 are delivered by providers of tracking images. It should be noted that in this statistics, we consider that all beacons found and also identical ones can meet on different pages, so the total number of beacons is not equal to the number of unique beacons.

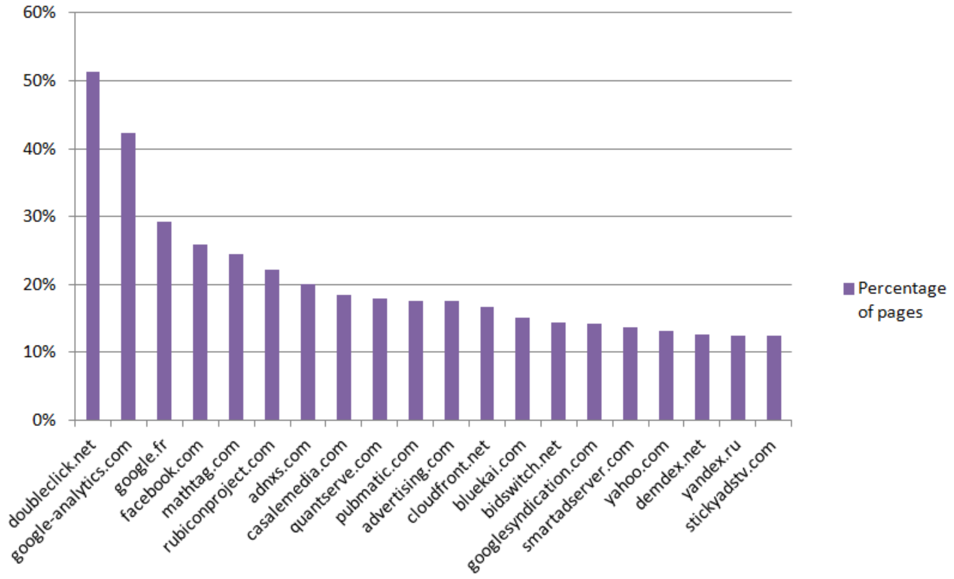


FIGURE 7 IMAGE DOMAINS – PERCENTAGE OF PAGES (TOTAL PAGES COUNT IS 124 214)

In Figure 7, it is demonstrated how many unique pages are served by providers of tracking images. Therefore there is at least one beacon image on such page.

In Figure 8, we can see how many unique domains are served by providers of beacon images. Thereby there is at least one page with at least one beacon on such domain.

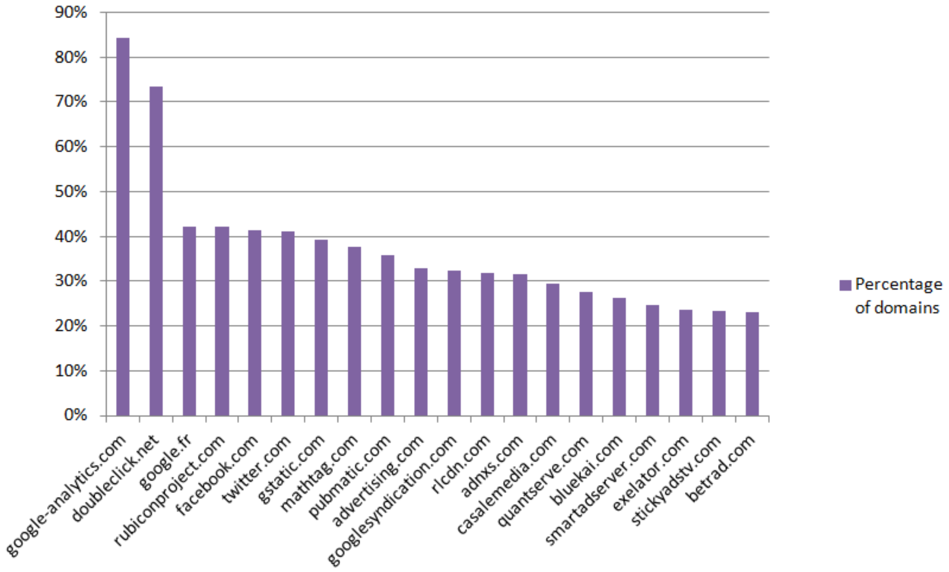


FIGURE 8 IMAGE DOMAINS – COUNT OF DOMAINS (TOTAL DOMAINS COUNT IS 800)

These plots demonstrate what companies provide web beacon tracking and prove that beacon images are the widespread form of web tracking.

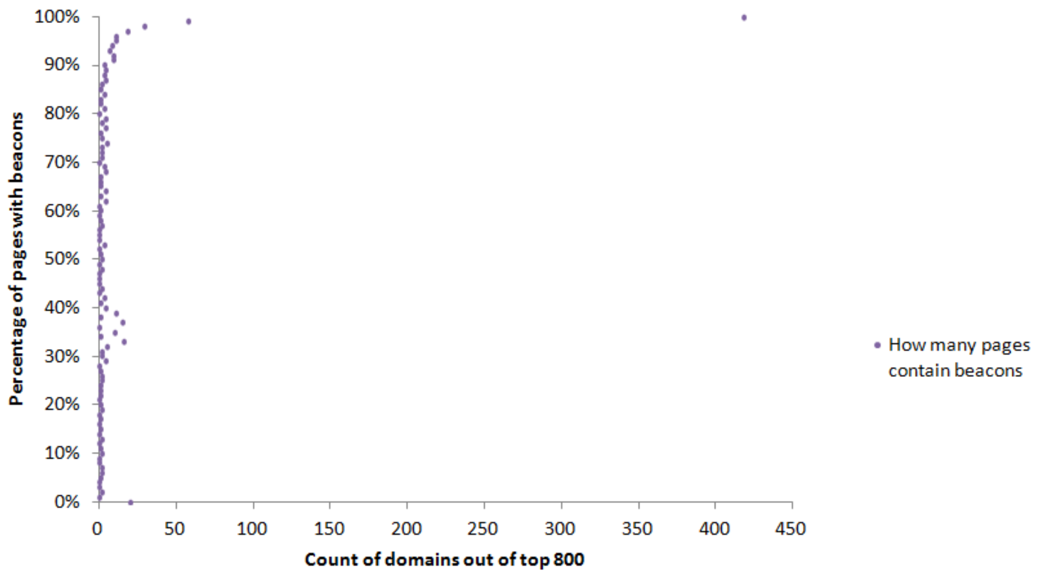


FIGURE 9 COUNT OF DOMAINS AND PERCENTAGE OF PAGES CONTAING BEACONS

The plot in Figure 9 shows the number of domains (out of top 800) and the percentage of pages that have at least one beacon (ratio to the total number of pages that have 3rd Party images).

Now when we know the top providers of beacons (Figure 8) it is interesting to know how these sites represent themselves for better understanding why they use web tracking technologies. Description of trackers can be found on website[3] of privacy tool that protects from behavioral ads and companies tracking users in Web.

Beacon domain	Category
google-analytics.com	Web analytics
doubleclick.net	Advertising
google.fr	Search Engine
rubiconproject.com	Advertising
facebook.com	Social Network
twitter.com	Social Network
gstatic.com	Web analytics
mathtag.com	Advertising
pubmatic.com	Web analytics
advertising.com	Advertising
googlesyndication.com	Advertising
rlcdn.com	Web analytics
adnxs.com	Advertising
casalemedia.com	Advertising
quantserve.com	Advertising

[3] https://better.fyi/trackers/

bluekai.com	Web analytics
smartadserver.com	Advertising
exelator.com	Web analytics
stickyadstv.com	Advertising
betrad.com	Web analytics

TABLE 6 CATHEGORIES OF TRACKERS

As we can see in Table 6, the prevalent categories of the top 20 trackers are web analytics and advertising. But web beacons are also used by search engine Google and social networks Facebook and Twitter. These trackers are more interesting as many users are logged into social accounts so this tracking does not seem anonymous.

In the next section we look into the standard ways to track users on the example of Google and Facebook beacons.

FACEBOOK AND GOOGLE BEACONS

Facebook pixel is used for conversion tracking; everyone can create his own beacon from his account page[4] in Adverts Manager.

The percentage of domains that contain on their pages such kind of beacons is 41.25%, so Facebook pixels are widely spread.

In database there are 751 unique Facebook pixels that have been found on 59 023 pages in Web. There are different patterns of links that return such beacons (Table 7).

№	Pattern	Unique pixels	Pages
1	https://facebook.com/tr/?	643 (85.61%)	56 629 (95.94%)
2	https://facebook.com/tr?	78 (10.38%)	2 245 (3.80%)
3	https://facebook.com/tr/	2 (0.26%)	67 (0.113%)
4	https://facebook.com/tr/brandlift.php?	15 (1.99%)	66 (0.111%)
5	https://facebook.com/tr/spacer.gif?	11 (1.46%)	11 (0.018%)
6	https://facebook.com/tr/offsite_event.php?	4 (0.53%)	4 (0.006%)

TABLE 7 PATTERNS OF FACEBOOK PIXELS URLS

[4] https://en-gb.facebook.com/business/a/facebook-pixel

Pattern №1 and №3 are used when JavaScript is allowed whereas pattern №2 is used when JavaScript is disabled in a browser. Pattern №3 is used for POST requests and patterns №1 and №2 for GET requests.

Pattern №4 is used for branded advertising. Pattern №6 is outdated; it has been used in the previous version of Facebook Pixel – Conversion Pixel that is why it is not widely spread.

Finally pattern №5 has been found only on pages of CocCoc browser's domain[5]. Interestingly, these initially spacer.gif images have been used not for tracking purposes but for styling[6] and have become archaic after the adoption of Cascading Style Sheets. At the same time patterns №1 and №3 also appear on the domain of CocCoc browser.

The following experiment shows whether tracking with Facebook pixel is anonymous. For the sake of purity of the experiment we should clean all cookies in a browser before beginning. First, it is necessary to accept 3rd party cookies in the browser's settings. Second, we go to a page that contains Facebook pixel and we monitor in browser network profiler to see what requests and responses occur and to know what exact cookies are being set and sent to Facebook server.

After first request to Facebook Pixel in a response's header Set-Cookie contain name "fr" and a unique value. With next GET or POST requests to this beacon the browser sends cookies that have been set on previous step that contain only value of "fr". It seems like an anonymous statistics for web analytics purposes and it does not violate users' privacy.

Let's take a look what happen than user is authorized into his Facebook account:

1. In the browser are appeared other cookies which are related to specific Facebook user: datr, sb, c_user, xs, csm, pl, lu, p, presence;
2. The field "fr" is updated with a new value.
3. With next GET or POST requests from the page with Facebook Pixel the browser sends cookies with values of datr, sb, c_user, xs, csm, pl, lu, p, presence and fr.

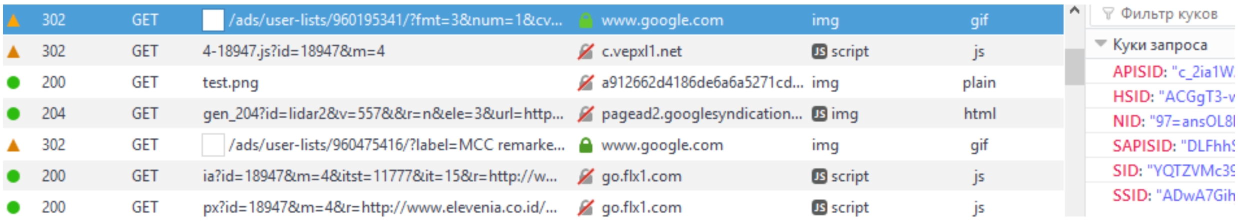
Hereby if a user is not logged into Facebook social network, a request from any advertising page (with Facebook pixel) sends one value in Cookie to Facebook related only to Facebook

[5] https://coccoc.com/en

[6] https://en.wikipedia.org/wiki/Spacer_GIF

pixel. But if this user is logged in, it starts sending also a user's identifier "c_user" and other values. Facebook knows what exactly a user visits what exactly a page so in this case it is not an anonymous statistics although they mention in Facebook Cookie Policy[7] that "We may also set and receive information stored in cookies from other domains".

Google tracks a user's browsing as well if a user is logged into his Gmail account. Before authorization in Google services, while requesting Google beacons a browser only sets a value "NID". But after authorization browser also sets and sends the values "SID", "HSID", "SSID", "APISID", "SAPISID" and "PAIDCONTENT" which have direct relation to users' accounts.



▲	302	GET	/ads/user-lists/960195341/?fmt=3&num=1&cv...	www.google.com	img	gif
▲	302	GET	4-18947.js?id=18947&m=4	c.vepxl1.net	JS script	js
●	200	GET	test.png	a912662d4186de6a6a5271cd...	img	plain
●	204	GET	gen_204?id=lidar2&v=557&&r=n&ele=3&url=http...	pagead2.googlesyndication...	JS img	html
▲	302	GET	/ads/user-lists/960475416/?label=MCC remarke...	www.google.com	img	gif
●	200	GET	ia?id=18947&m=4&itst=11777&it=15&r=http://w...	go.flx1.com	JS script	js
●	200	GET	px?id=18947&m=4&r=http://www.elevenia.co.id/...	go.flx1.com	JS script	js

Фильтр куков

Куки запроса

APISID: "c_2ia1W

HSID: "ACGgT3-v

NID: "97=ansOL8I

SAPISID: "DLFhh

SID: "YQTZVMc3

SSID: "ADwA7Gih

FIGURE 10 SENDING COOKIES VIA GOOGLE BEACON

Though modern browsers support blocking third-party cookies prohibiting their setting and sending at the same time. During experiments in the work it has been discovered that this option works appropriately in the browsers Safari 10.0.3, Firefox 51.0.1 and in Chrome 56.0.2924.87 thus preventing such tracking behavior.

[7] https://www.facebook.com/policies/cookies/

CONCLUSION

While users surfing the Internet, their website visits, choices and preferences are continuously being monitored by tracking companies. This information can be used not only for targeting advertisements but also for discrimination of users, for instance by providing varying prices for goods depending on user’s solvency.

In this work it is shown that web tracking via invisible web beacons is a widely spread form of tracking nowadays. Beacons may have non standard sizes and they can work with disabled JavaScript and cannot be avoided by Private browsing mode. Though there is the option "Don't accept 3rd party cookies" in modern browsers that prevents this tracking behavior but it is turned off by default in the most of them.

During learning the way Facebook beacon works it has not been discovered that they use anything aside from cookies for saving users’ identifiers but theoretically they can. In this case if websites starts keeping this information in other places for example in HTML Local Storage, blocking third-party cookies would not be helpful against tracking via beacons.

In a further work it is necessary to find a way to prevent this kind of web tracking because it is not obvious even for knowledgeable users and may infringe their privacy.

BIBLIOGRAPHY

1. F. Roesner, T. Kohno, and D. Wetherall. Detecting and Defending Against Third-Party Tracking on the Web. In *WWW*, 2012.
2. H. Yu. Do Not Track: Not as Simple as it Sounds, Aug. 2010. https://freedom-to-tinker.com/blog/harlanyu/donot-track-not-simple-it-sounds
3. C. Jackson, A. Bortz, D. Boneh, and J. C. Mitchell. Protecting browser state from web privacy attacks. In *WWW*, 2006
4. G. Acar, C. Eubank, S. Englehardt, M. Juarez, A. Narayanan, C. Diaz. The Web Never Forgets: Persistent Tracking Mechanisms in the Wild. In *Proceedings of CCS 2014*, Nov. 2014.
5. M. Ayenson, D. J. Wambach, A. Soltani, N. Good, and C. J. Hoffnagle. Flash Cookies and Privacy II: Now with HTML5 and ETag Respawning. *Social Science Research Network Working Paper Series*, 2011.
6. T.-F. Yen, Y. Xie, F. Yu, R. P. Yu, and M. Abadi. Host fingerprinting and tracking on the web: Privacy and security implications. In *NDSS*, 2012.
7. G. Aggrawal, E. Bursztein, C. Jackson, and D. Boneh. An analysis of private browsing modes in modern browsers. In *Usenix Security Symposium*, 2010.
8. Jay Goldman. Facebook Cookbook: Building Applications to Grow Your Facebook Empire. *O'Reilly Media*, 2008.