Analysis of OpenX-Publishers Cooperation

Lukasz Olejnik[1] and Claude Castelluccia[1]

[1] INRIA, Grenoble, France, Email: lukasz.olejnik@inria.fr
[2] INRIA, Grenoble, France, Email: claude.castelluccia@inria.fr

Abstract. Real-Time Bidding is a protocol enabling the serving of advertisements. It involves Ad Exchanges, bidders and publishers. In this note, we report the findings of cooperation between OpenX Ad Exchange and selected publishers. The setting has potentially important implications for Web users privacy and security. For example, Web browser mechanisms responsible for blocking third-party cookies are rendered ineffective.

1 Introduction

Real-Time Bidding (RTB) [12, 8] is already a vibrant and ubiquitous technology allowing the display of advertisements to users based on decisions made in real time. When a user visits a Web site which supports RTB, the RTB system (Ad Exchange, AdX) holds an auction: it sends *bid requests* to its bidders. The auction's bidders submit their bids and the winner displays its advertisement, later issuing a payment for this benefit. The advertisement is displayed on the publisher's Web site. RTB is thus a system composed of three primary, separate parties.

Bidders make decisions based on data obtained from Ad Exchanges during the auction phase. This data usually contains information on the user, for example his *currently-visited site*, inferred gender or ethnicity. Thus users' private data are being exchanged.

RTB is actively used to build detailed profiles of users [21]. In December 2013, eMarketer was predicting that the market share will reach 29% in 2017, with $\$9B$ ad spending devoted to RTB [9]. In the meantime, RTB massively grows in certain markets such as China, with 437% increase from Q3 to Q4 (2013). This is driven by the key Chinese players: Taobao, Tencent, Sina, and Baidu [4]. Real-Time Bidding's disruptive potential is exemplified by its possible direct influence on the 2016 US presidential elections, where users will be targeted by their physical locations, political affiliations, age in addition to much detailed information such as ownership of a gun [6].

During our analyses of privacy and transparency in Real-Time Bidding, we detected that the specific setting employed by OpenX Ad Exchange results in the evasion of third-party cookies blocking. We enlist the consequences of the described setting.

1. In case of browsers which block third-party cookies ("*blocking cookies from not visited sites*"), the cookies are being set, effectively bypassing configuration of these browsers.
2. In case of certain Web sites, potential user-related cookies are being leaked to OpenX Ad Exchange. This may result in an informational advantage, as there are possibilities of storing the user-related data such as e-mails, directly in the cookies (e.g. [27]). Among the information commonly stored in cookies are also session identifiers. Thus, the analyzed setting has potential security implications.

We acknowledge an example of relatively recent issues where companies were using specific settings to purposefully set cookies for users of Safari browser. In this case, it has brought the attention of a regulatory body, the US Federal Trade Commission [10, 20].

The technical details behind the scheme utilized to set cookies in Safari browser [1] were, however, entirely different to those we mention in this work. The difference is that in those cases, a specially crafted JavaScript code was in place. In this work, we describe the use of DNS aliasing by OpenX which might lead to leaks of cookies from the publishers' sites. The importance of this finding is heightened by the fact that OpenX is an Ad Exchange and maintains Real-Time Bidding auctions.

2 Background

2.1 Related Work

Browsers can be tracked by many means, as Jackson et al. analyze in [14]. Current browsers are very limited when it comes to compartmentalization. In particular, it is not straight-forward to block third-party cookies [14].

The so-called *hidden third-party* model where a third-party site *X.com* is granted permission to operate a subdomain of a publisher *Y.com* is a well known problem and was mentioned by Krishnamurthy et al. [17, 16] and Wills [27]. The resulting setting where *X.com* controls a subdomain (*domain.Y.com*) of *Y.com* may bring consequences to users. For example, if *Y.com* maintains user accounts or stores other user-related sensitive data in the browser cookies with inappropriate scope, a leak to *X.com* might occur when the user visits *Y.com*. An interesting example is the one where *Y.com* stores user's e-mail address directly in the cookies [27]. In this case, *X.com* can obtain access to this data.

Meyer et al. [19] thoroughly describe the problem of first-party Web sites allowing the tracking of their users via third-party scripts. The authors approach the problem also from a policy point of view and note that the US Federal Trade Commission is recently involved in Web tracking regulations. In fact, there are recent cases where a party trying to evade browser's third-party cookie blocking mechanisms [24] drew US Federal Trade Commission's sanctions.

There are many possibilities of assigning a unique identifier to Web users, with a tracking purpose. Browser cookies, Flash cookies (also called *Local Shared*

Objects), ETags and others can be used, or even combined to make the removal of user identifiers more difficult. For example, if the user decides to clear all of the browser cookies, a tracker could reconstruct them out of Flash cookies. As a result, the identifier becomes more persistent: a *zombie cookie* (or *evercookie* [15]).

Until recently, it was not obvious how to remove a Flash cookie, a functionality made available only in 2011 [5]. For example, records of the regeneration of the previously-removed browser cookies out of Flash cookies exist [25]. But this reproducing ability is not limited to Flash cookies, as ETag feature of the browser cache [2] and others can also be used. In fact, numerous companies were found to be applying these practices [11, 7, 22].

Another recent risk is related to cookie synchronization: the mechanism of matching the cookies of two separate entities enables their reconstruction. For example, domains *X.com* and *Y.com* can match their cookies for a particular user. If this user later removes a cookie of *X.com*, and then during Web browsing encounters a site with *X.com*'s scripts, it is possible that *X.com* will regenerate its cookies (the ones removed by the user) using the previous mapping granted by *Y.com*. This setting was found to be employed by (e.g.) Microsoft [18]. In the Web economy and particularly in RTB where the usual tracking and identification means are browser cookies [3], cookie matching is routinely used [21]. Ad Exchanges set their cookies in the users' browsers, and then construct user identifiers which are being sent to their bidders in bid requests (part of RTB auction process).

The Web community sought precautions against tracking. Browser extensions such as AdBlock Plus or Ghostery can grant Web users more control over the elements (ads, analytics, etc.) they wish to be exposed to.

2.2 Real-Time Bidding and Price Notification

When a user visits a *publisher*'s Web site which supports RTB, the *RTB system* (*Advertising Exchange*, *AdX*) holds an *auction* for this *ad impression*. The auctions are sealed-bids according to Vickrey principles [26], where the second largest price (increased by a constant) is to be paid by the winner. The auction participants are composed of *bidders* who bid on behalf of the advertisers and/or perhaps different Ad Exchanges.

During the auction, *RTB systems* send *bid requests* to the *bidders*. These requests can contain information about the user, such as the *visited site*, the *IP address* (or parts of it) and even the inferred *ethnicity*[3] and *income*; therefore the user's private data are being transferred to the participating parties. Participants of the auction appraise the received data and submit their bids to the auction holder. The whole process typically takes less than 100 milliseconds. The winner's advertisement is then displayed in the user's browser.

[3] OpenX can provide the following ethnicity information: *African American*, *Asian*, *Hispanic*, *White* and *Other* in its bid request

The user's browser requests the advertisement via a standard HTTP GET request of an URL. The request is executed by a script present in the winning bidder's ad snippet, supplied by the RTB. That ad snippet contains elements such as HTML tags, responsible for performing a request. This request's URL very often carry a price notification. The price notification is meant to inform the winner about the monetary value to be paid for the displaying of this advertisement, the *winning bid*. The URL is therefore a HTTP request of the form: *http://bidder.com?price=encrypted-price*, where *price* is the name of the price-containing parameter, and *encrypted-price* is the actual price notification. The notification is very often in an encrypted form, conforming to the industry standard, the so-called *encrypted price* [13]. By analyzing the flows of messages with encrypted prices conforming to the standard, it is possible to obtain a list of bidders utilizing RTB systems. Figure 1 shows a schematic diagram of interactions between the involved parties.

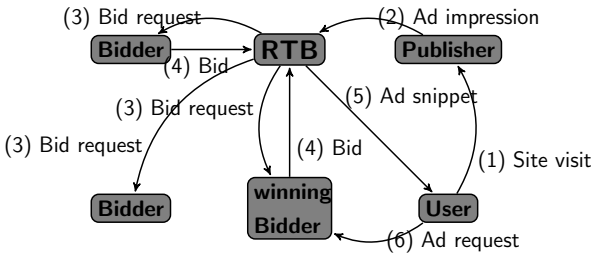

Fig. 1: Parties involved in RTB.

In this work, we analyze interactions between Ad Exchanges and bidders, as well as Ad Exchanges and publishers. In general, the flow of sending bid requests is as follows: *Ad Exchange* $\rightarrow$ *bidder*. Bidders then submit their bids and the winning bidder's ad snippet is transmitted to the user's browser and displayed on the Web site *example.com* that the user is visiting. This is performed by an Ad Exchange, which is serving the content responsible for displaying the winning bidder's ad. For example, if a bidder App Nexus (*adnxs.com*) wins an auction at Doubleclick Ad Exchange (*doubleclick.net*), a schematic flow might be: *doubleclick.net* $\rightarrow$ *example.com*. The ad is then typically displayed on the visited Web site.

In one case, we detected examples of an unusual setting between the Ad Exchange and Publishers. This work is devoted solely to the study of this setting.

3 OpenX-Publishers Cooperation

3.1 General Case

In RTB, publishers' sites such as *example.com* usually include third-party scripts to provide ad impressions and display ads. Technically this is often done by the

use of a HTML `iframe` tag. An example domain name of such scripts can be *doubleclick.net*, in case of Doubleclick. Ad Exchanges usually set specific tracking cookies in the users' browsers. For instance, Figure 2 shows that when the user visits *example.com*, a request to *doubleclick.net* is made; during this request, the user's cookie controlled by Doubleclick is sent. The winning bidder's (Criteo in this case) ad snippet is then served and the procedure of serving ads can be initiated.

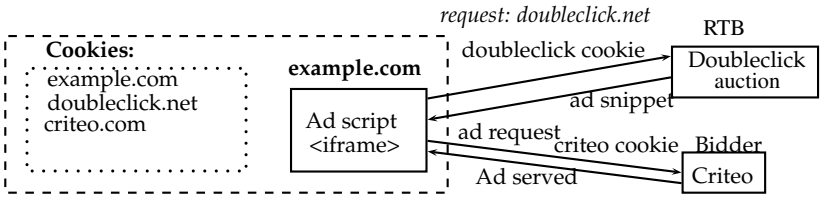

Fig. 2: Inclusion of ad snippets on publishers' sites (General case).

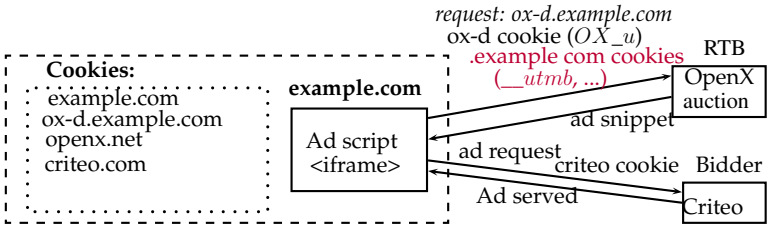

Fig. 3: Inclusion of ad snippets on publishers' sites (OpenX case).

3.2 OpenX Case

We noticed a puzzling cooperation between one Ad Exchange and certain publishers. In this setting, the publisher *example.com* includes an `iframe` referring to the script from the domain *ox-d.example.com*. This subdomain, supposedly belonging to the visited site (*example.com*) is in reality a DNS alias.[4] For example in case of *dailyherald.com*, *ox-d.dailyherald.com* is an alias for *dailyherald-d3.openxenterprise.com*. This *hidden third-party* setting is an example of DNS aliasing [17, 16, 27]. The actual server is controlled by a third-party, OpenX Ad Exchange.[5]

[4] For a comprehensive introduction to DNS aliasing, please refer to `http://en.wikipedia.org/wiki/CNAME_record`

[5] *http://openx.com*

The consequences of this scheme for browser cookies are especially interesting. The domain *ox-d.example.com* sets a cookie with a unique user identifier, *OX_u* (example in Table 1). This means that when the user's browser performs a request to this host, the cookie is included in this request's headers. However, very often the first-party domain name, *example.com*, sets its own cookies as well. If these cookies have a *broad scope* (for example *Domain=.example.com* [3]), they are leaked to *ox-d.example.com*, which is a site operated by OpenX, an external entity. As shown on Figure 3, not only legitimate cookies, controlled by OpenX (belonging to *ox-d.example.com*), are transferred to the third-party server, but other first-party cookies unrelated to OpenX are sent as well.

This means that in this setting, OpenX could have access to the cookies of *example.com*, due to their scope. As a matter of fact, we detected that cookies are leaking to OpenX systems on several of the analyzed Web sites. Examples of the affected Web sites are *dailyherald.com* (leak to *ox-d.dailyherald.com*) and *popcrunch.com* (leak to *ox-d.popcrunch.com*). Whenever a user visits these sites, certain cookies are leaked to OpenX, depending on the cookie scoping.

Table 1 shows an example: *OX_u* is the per-user cookie of OpenX, while *__utmz* is a cookie related to Google Analytics. Its scope is *.popcrunch.com*, which means this cookie is included during a HTTP request to any subdomain of *popcrunch.com*. Therefore, it is also sent to *ox-d.popcrunch.com*, a hostname controlled by OpenX. We will detail the privacy implication of this leakage in Section 4.

Name	**Value**	**Scope**
__utmz	236312704.1392366853.1.1.utmccn=(direct)— *[...]*	.popcrunch.com
OX_u	195e9f99-9b18-0991-06a2-98172b0d3651_m_1385039804	ox-d.popcrunch.com

Table 1: OpenX cookies (OX_u) and Google Analytics (__utmz) one. In this case __utmz leaks to OpenX due to cookie scoping.

4 Privacy Analysis of OpenX Setting

4.1 Privacy Measurement Setting

We crawled $1M$ Alexa[6] sites and searched for requests of the form *ox-d.example.com*, where *example.com* is the address of the visited site. We identified 127 sites. When performing a name resolution, all such hosts turned out to be operated by OpenX. Using PhantomJS browser, we subsequently visited each of these 127 sites and saved all the related cookies. We detected that OpenX's cookie was set on about 20% of the sites (i.e. 26 sites).

We then analyzed cookies for each of these 26 Web sites *example.com*, in order to verify if cookie leaks take place. In case of 81% of these sites, we detected

[6] `http://www.alexa.com`

the leakage of cookies not belonging to OpenX. Those cookies commonly had a broad scope set. We found that *Google Analytics* cookies (e.g. `__utma`) were leaked to OpenX servers in 70% of these sites. For example, the site *zam.com* leaked three cookies, two of them being related to Google Analytics.

4.2 Consequences: Third-Party Cookie Blocking Circumvention

Cookies are often directly related to authentication in Web systems [23]. Consequently, this leak has both security and privacy implications [16, 27]. Krishnamurthy and Wills discussed such possibilities in [16]. Wills mentioned the risks of these settings where first-party site stores sensitive data such as user's e-mail address, in the cookies [27]. In the case we analyzed, the problem might even be more complicated due to the fact that OpenX is an Ad Exchange and has real-time bidders who receive bid requests with information on the user.

By leveraging this close RTB-publisher collaboration, it is possible to evade host-based blacklists of advertising and third-party tracker-blocking browser extensions, such as Ghostery. Moreover, this setting also effectively serves as *a work-around against blocking of third-party cookies*, a mechanism used by Safari browser and currently considered for inclusion to Firefox. We believe this might be a primary motivation in the encountered cases.

We performed a test with Firefox 26. We enabled *blocking of cookies from the sites the users did not visit* (third-party cookies). We detected that when visiting *popcrunch.com*, *OX_u* cookies were still set by *ox-d.popcrunch.com*, a consequence of how third-party cookie blocking works in Firefox. We also installed Ghostery and enabled its blocking mode. Requests to *ox-d.popcrunch.com* were still executed and OpenX's cookie was set.

4.3 Cookie Matching Potential

One of the potential consequences of this setting is that OpenX could create a custom Cookie Matching scheme [21], where cookies set by two different entities are mapped. OpenX could match their *user id* cookies with e.g., Google Analytics cookies belonging to these Web sites. As a result, OpenX could track the visitors of these Web sites based on the per-site cookies of Google Analytics. Example scenario could arise when the user removed OpenX cookies (or when they expired) but left the ones belonging to Google Analytics intact. OpenX could then regenerate the user's profile using the unchanged Google Analytics cookie. In the case of example from Table 1, we manually verified that OpenX was not reproducing its tracking cookie out of Google Analytics cookies, after its removal. But linking the profile with the new cookie is still theoretically possible.

4.4 Countermeasures

Deployment of the analyzed setting complicates the blocking of third-party cookies. However, there are still viable options. One possible solution is AdBlock

Plus browser extension. ABP can be used to block the setting. The default filter list provide rules enabling the blocking of those requests. For example, the rule `ox-d.*^auid=` matches against requests to `http://ox-d.example.com/auid=...`. This would effectively block all requests to these domains. According to ABP's Web site,[7] the extension received over $200M$ downloads. Obviously, the users without the extension are left unprotected.

5 Conclusion

In the case we analyzed in this work, a scheme of DNS aliasing is used. One of the consequences is that users' cookies of first-party Web sites are being sent to a technically third-party site. We note that the employed approach also circumvents third-party cookie blocking.

Recently, regulatory attention was drew in a corresponding case [10] to the one we describe in this work. Google has agreed on a settlement related to the evasion of Safari's blocking of third-party cookies [20].

Acknowledgments. We thank the anonymous reviewers for their comments.

References

1. J. Angwin and J. Valentino-Devries. Google's iPhone tracking. The Wall Street Journal. `http://online.wsj.com/news/articles/SB10001424052970204880404577225380456599176`, 2012.
2. M. Ayenson, D. J. Wambach, A. Soltani, N. Good, and C. J. Hoofnagle. Flash cookies and privacy ii: Now with html5 and etag respawning. World Wide Web Internet And Web Information Systems, 2011.
3. A. Barth. HTTP State Management Mechanism. `http://tools.ietf.org/html/rfc6265`, 2011. RFC 6265.
4. Brandscreen. Brandscreen Q4 2013 real-time media insights report. `http://www.brandscreen.com/en/blog/brandscreen-q4-2013-real-time-media-insights-report`, 2013.
5. M. Brinkmann. A close look at adobe flash player 10.3 beta. `http://www.ghacks.net/2011/03/08/a-close-look-at-adobe-flash-player-10-3-beta/`, 2010.
6. Business Insider. How real-time bidding will influence the 2016 election — and change Don Draper's job. `http://www.businessinsider.com/sc/real-time-bidding-helps-decide-elections-2014-1`, 14 January, 2014.
7. J. Cheng. Zombie cookie wars: evil tracking api meant to "raise awareness". `http://arstechnica.com/business/2010/09/evercookie-escalates-the-zombie-cookie-war-by-raising-awareness/`, 2010.
8. DoubleClick. Doubleclick ad exchange real-time bidding protocol. `https://developers.google.com/ad-exchange/rtb/`.

[7] `http://adblockplus.org`, accessed in May 2014

9. eMarketer. Advertisers continue rapid adoption of programmatic buying. http://www.emarketer.com/Article/Advertisers-Continue-Rapid-Adoption-of-Programmatic-Buying/1010414, 2013.
10. FTC. Google will pay $22.5 million to settle ftc charges it misrepresented privacy assurances to users of apple's safari internet browser. http://www.ftc.gov/news-events/press-releases/2012/08/google-will-pay-225-million-settle-ftc-charges-it-misrepresented, 2012.
11. A. Gonsalves. Company bypasses cookie-deleting consumers. http://www.informationweek.com/company-bypasses-cookie-deleting-consumers/d/d-id/1031518?, 2005.
12. Google. The arrival of RTB. http://static.googleusercontent.com/external_content/untrusted_dlcp/www.google.com/pl//doubleclick/pdfs/Google-White-Paper-The-Arrival-of-Real-Time-Bidding-July-2011.pdf.
13. Google. Decrypting price confirmations. https://developers.google.com/ad-exchange/rtb/response-guide/decrypt-price.
14. C. Jackson, A. Bortz, D. Boneh, and J. C. Mitchell. Protecting browser state from web privacy attacks. In Proceedings of the 15th international conference on World Wide Web, pages 737–744. ACM, 2006.
15. S. Kamkar. Evercookie. http://samy.pl/evercookie, 2010.
16. B. Krishnamurthy and C. Wills. Privacy diffusion on the web: a longitudinal perspective. In Proceedings of the 18th international conference on World wide web, pages 541–550. ACM, 2009.
17. B. Krishnamurthy and C. E. Wills. Generating a privacy footprint on the internet. In Proceedings of the 6th ACM SIGCOMM Conference on Internet Measurement, IMC '06, pages 65–70, New York, NY, USA, 2006. ACM.
18. J. Mayer. Tracking the trackers: Microsoft advertising. http://cyberlaw.stanford.edu/node/6715, 2011.
19. J. R. Mayer and J. C. Mitchell. Third-party web tracking: Policy and technology. In Security and Privacy (SP), 2012 IEEE Symposium on, pages 413–427. IEEE, 2012.
20. C. C. MILLER. Google to pay $17 million to settle privacy case. http://www.nytimes.com/2013/11/19/technology/google-to-pay-17-million-to-settle-privacy-case.html?_r=1, 2013.
21. L. Olejnik, M.-D. Tran, and C. Castelluccia. Selling off user privacy at auction. In In Proceedings of the 21st Annual Network and Distributed System Security Symposium (NDSS). The Internet Society, 2014.
22. Out-law.com. Web users sue companies claiming use of flash cookies is a hack. http://www.out-law.com/page-11318, 2010.
23. OWASP. Owasp periodic table of vulnerabilities - cookie theft/session hijacking. https://www.owasp.org/index.php/OWASP_Periodic_Table_of_Vulnerabilities_-_Cookie_Theft/Session_Hijacking, 2013.
24. R. Singel. Google busted with hand in safari-browser cookie jar. http://www.wired.com/2012/02/google-safari-browser-cookie/, 2012.
25. A. Soltani, S. Canty, Q. Mayo, L. Thomas, and C. J. Hoofnagle. Flash cookies and privacy. In AAAI Spring Symposium: Intelligent Information Privacy Management, 2010.
26. W. Vickrey. Counterspeculation, auctions, and competitive sealed tenders. The Journal of finance, 16(1):8–37, 1961.

27. C. E. Wills. Identifying and preventing conditions for web privacy leakage. In Proceedings of the W3C Workshop on Web Tracking and User Privacy, 2011.