David Gugelmann*, Markus Happe, Bernhard Ager, and Vincent Lenders

An Automated Approach for Complementing Ad Blockers' Blacklists

Abstract: Privacy in the Web has become a major concern resulting in the popular use of various tools for blocking tracking services. Most of these tools rely on manually maintained blacklists, which need to be kept up-to-date to protect Web users' privacy efficiently. It is challenging to keep pace with today's quickly evolving advertisement and analytics landscape. In order to support blacklist maintainers with this task, we identify a set of Web traffic features for identifying privacy-intrusive services. Based on these features, we develop an automatic approach that learns the properties of advertisement and analytics services listed by existing blacklists and proposes new services for inclusion on blacklists. We evaluate our technique on real traffic traces of a campus network and find in the order of 200 new privacy-intrusive Web services that are not listed by the most popular Firefox plug-in Adblock Plus. The proposed Web traffic features are easy to derive, allowing a distributed implementation of our approach.

Keywords: Privacy; Web; tracking; advertisement; analytics; blacklist; HTTP; network measurement

DOI 10.1515/popets-2015-0018
Received 2015-02-15; revised 2015-05-15; accepted 2015-05-15.

1 Introduction

Advertisement and analytics Web services are embedded in many Web sites [48] and track the user behavior across sites to generate exhaustive user profiles. They use the gathered user data to estimate the gender, age, credit worthiness, health state, etc. of a person [3, 16, 25, 28, 39]. User profiles are then used for customized advertisements or simply sold [18]. Therefore, in principal any party can get access to the collected personal information. This can have severe consequences for individuals [25] and organizations. For example, individuals can experience higher prices based on their profile [33], can be rejected by an insurance company [50], or not be invited to a job interview [3, 39].

Corporations face the problem that third-party Web services can learn about "hot topics" in organizations. Web services collecting corporate interest profiles can use their knowledge to predict business processes, including but not limited to strategic trading on the stock market, or they could simply sell uncovered business secrets to the organization's competitors. Assuming that an organization uses a fixed public IP address range for its Web access, privacy-intrusive third parties do not even need user identifiers because they can identify organizations based on the source IP addresses of requests [27]. This puts also third parties that are no trackers per se, e.g., content distribution networks or services hosting popular Web utilities, in the position to track the Web activities of an organization if no mitigation measures are in place.

It is a major challenge to protect individuals and corporations against privacy-intrusive Web services. Prevention measures have to find a good trade-off between privacy protection and usability. Rigorous default-block policies, e.g., not allowing any JavaScripts, are efficient [43] but break functionality on many Web sites. In order to maintain the functionalities of Web sites, default-block policies need fine-grained exception rules, which many users are unwilling to create. As a consequence, most privacy-protecting browser plug-ins (such as Adblock [40], Adblock Plus [41], or Ghostery [42]), as well as corporate Web security gateways operate with a default-accept policy and rely on blacklists to identify privacy-intrusive HTTP requests to be blocked. This is much more comfortable for users as the maintainers of the blacklists usually try to ensure that Web site functionality remains intact. However this approach also comes with drawbacks:

– The advertisement and analytics sector is a very dynamic, fast-growing business sector with existing trackers disappearing and new trackers coming. Manually maintaining blacklists requires consider-

*Corresponding Author: David Gugelmann: ETH Zurich, Switzerland, E-mail: gugelmann@tik.ee.ethz.ch
Markus Happe: ETH Zurich, Switzerland, E-mail: mhappe@tik.ee.ethz.ch
Bernhard Ager: ETH Zurich, Switzerland, E-mail: bager@tik.ee.ethz.ch
Vincent Lenders: armasuisse, Switzerland, E-mail: vincent.lenders@armasuisse.ch

able effort to keep up with changes in the tracking landscape and blacklists risk to miss new privacy-intrusive services.
– It is often not clear which criteria blacklist maintainers use to decide if a service should be blacklisted. For example, Adblock Plus has received money from multiple large Web services to be white-listed [9].

As a step towards addressing these problems, we propose a novel approach for identifying new privacy-intrusive services based on HTTP traffic statistics. Our work can be applied to distributed collected data as well as in a local network: *(i)* to identify new blacklist candidates, blacklist maintainers can use our methodology in a *global* fashion by applying the proposed classifier to traffic statistics downloaded from a crowd-sourced database; *(ii)* companies can apply our model on their *local* HTTP traffic to verify that a provided blacklist indeed blocks relevant services and enhance provided blacklists according to their preferences.

This paper provides the following contributions:

1. A comprehensive analysis of HTTP traffic statistics that are suitable to identify privacy-intrusive services.
2. A classifier for identifying privacy-intrusive Web services. The classification of a 24 h trace of a campus network with 15 k clients showed a precision as well as recall of around 84% when classifying popular Web services.
3. An exhaustive evaluation of our automated approach. We have applied our proposed classifier to complement the manually maintained blacklists of a popular privacy-protecting browser extension. By learning using Adblock Plus blacklists and HTTP traffic traces, both from August 2013, our classifier identified >400 new Web services that exhibit typical characteristics of privacy-intrusive services. Around 200 of these services are advertisement and analytics services in the traditional sense, while most remaining services are in the gray area, i.e., it depends on user or organization preferences if corresponding services should be regarded as a privacy risk or not.[1] As of February 2015, the Adblock Plus community indeed identified 70 of these services too and added them to their blacklists.

We believe that especially popular privacy-protecting browser plugins such as Adblock, Adblock Plus, or Ghostery can benefit from our automatic identification process to identify new privacy-intrusive Web services.

Our work is structured as follows: Section 2 gives background information on user profiling and corresponding protection measures. Section 3 presents our experimental design and Section 4 our dataset. We analyze properties of blacklisted services in Section 5. In Section 6, we develop a classifier. We present our experimental results in Section 7 and discuss a global and a local usage scenario of our technique in Section 8. Finally, Section 9 presents related work, and Section 10 concludes our study.

2 Background

In this section, we give a brief introduction to the two main mechanisms causing information on visited Web sites to be leaked to third parties (Section 2.1) and discuss prevention measures (Section 2.2). A more complete overview of techniques, policies, protective measures, and risks involved in Web tracking can be found in Mayer and Mitchell's survey paper [32].

2.1 User profiling by third-party Web services

Modern Web pages consist of dozens to hundreds of embedded objects [8, 45] which have to be fetched before a Web page can be displayed, such as images, videos, or JavaScript objects. Many of these objects are not loaded from the visited Web service (the *first party*) but from other Web services (*third parties*) [32]. For example, images and videos are often fetched from dedicated content distribution networks (CDN) with high-bandwidth links, and advertisements or analytics objects from companies specialized on advertising. Third parties can typically infer user behavior via two main mechanisms: with the HTTP Referer header, or by uploading information via embedded scripts.

Referer header

HTTP requests contain a Referer header which identifies the Web page linking to or embedding the requested resource of the third-party Web service [17]. For exam-

1 For example, some users extensively use social services, while others try hard to avoid revealing any information to social networks.

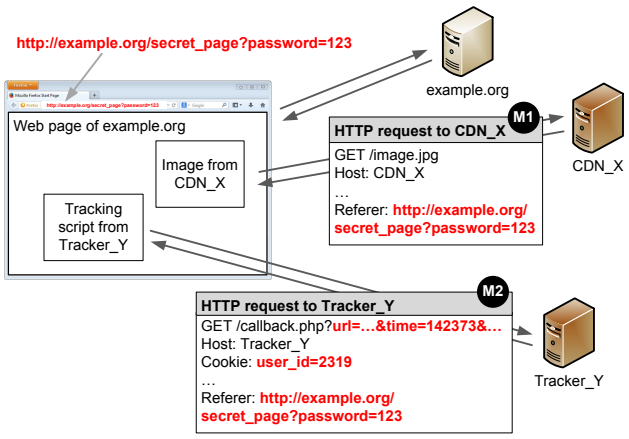


Fig. 1. HTTP data flow between a Web page on *example.org* and two third parties, a content delivery network (*CDN_X*), and a tracking service (*Tracker_Y*). The URL of the currently visited Web page is transferred to any third party over the Referer field in the HTTP header (mechanism **M1**). Most tracking services additionally employ scripts to collect data in the user's browser and send them back using HTTP request parameters (mechanism **M2**). Cookies, transmitted with every HTTP request as shown in M2, are the basic mechanism for re-identifying users.

ple, request M1 in Figure 1, shows how a content distribution network (CDN) providing an image embedded in a Web site can learn the address of the currently loaded Web page through the Referer header. Sending Referer headers is the default behavior on all modern browsers, therefore any third party (including but not limited to advertisement and analytics) can *passively* learn which Web pages are visited by a user.

Embedded scripts

Many third parties request the developer of the first-party Web site to embed a JavaScript program. The third party's program executes inside the user's browser with the same permissions as the first party's domain. This enables the third party to collect detailed information on the user's browsing activity, e.g., the position of the mouse pointer on the screen [54] and the browsing history [35], or to reconstruct unique user identifiers [2]. The collected information is then uploaded to the third party's Web server, see example M2 in Figure 1. Embedding a script requires *active* involvement in the Web site delivery process, thus these scripts are typically only used by advertisement and analytics services.

2.2 Protection against tracking

The simplicity of techniques allowing to collect information about a user's browsing behavior begs the question if there aren't similarly simple counter-measures. In particular, if disabling the corresponding functionality at the user's browser could fix the problem. Of course, Referer headers can be suppressed. Unfortunately removing the Referer header simplifies cross-site request forgery attacks [17]. In addition, some Web sites check Referer headers to prevent others from deep linking their Web pages, or to prevent bandwidth stealing. Similarly, the execution of JavaScript can be blocked [43]. However, blocking script execution has severe drawbacks on Web site functionality. Consequently, successful application of Referer header removal and script blocking require a user to perform elaborate control over which sites can receive Referer headers, respectively which scripts may be executed [1, 43].

A more user friendly approach appears to be the targeted blocking of requests to advertisement and analytics services, such as employed by Adblock Plus [41]. This targeted approach is implemented with centrally maintained blacklists, in case of Adblock Plus by a set of community maintained blacklists.

3 Experimental design

In this section, we present the objective of our study in Section 3.1 and outline the applied procedure in Section 3.2. Further, we introduce the analyzed traffic features in Section 3.3 and discuss the granularity at which we identify Web services in Section 3.4.

3.1 Objective

Our aim is to *simplify the maintenance* of blacklists of tracking services. Towards this goal, we present a machine learning approach for the automated identification of privacy-intrusive services, i.e., we classify Web services into two classes: *(i) privacy-intrusive* and *(ii) not privacy-intrusive (other)*. While advertisement and analytics services are the classic example of privacy-intrusive services that many users and organizations want to block, there are also services that are in a gray area because they provide benefits to some people. As such individual preferences are difficult to generalize, we focus on advertisement and analytics services while

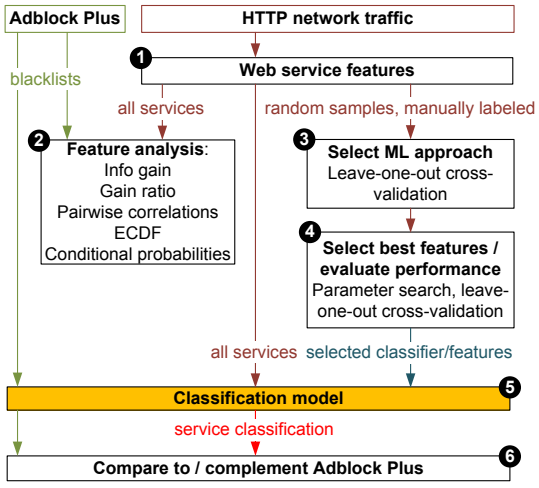


Fig. 2. Overview of our study.

developing our method. Our work is guided by the following goals:

1. We aim to find a large amount of privacy-intrusive services with minimal manual effort.
2. We prefer features with limited privacy impact to facilitate a collaborative detection approach.
3. We favor a lightweight process using features that are easy to derive to enable a quick adaption to the fast-changing service landscape.

3.2 Procedure

Our work is based on the observation that many advertisement and analytics services show *different traffic properties* than regular services. In order to find robust traffic features and training methods that can be used to classify privacy-intrusive services, we perform a study whose procedure is depicted in Figure 2 and outlined in the following.

Feature extraction and analysis

We first introduce traffic features and conduct an initial analysis.

1. We develop Web service traffic features for distinguishing privacy-intrusive from other services in Section 3.3. The granularity at which we identify Web services is discussed in Section 3.4.
2. In order to gain deeper insights into typical traffic patterns, we extract real-world feature statistics from HTTP traffic observed in a large campus network (see Section 4.1) and classify the Web services using existing blacklists (see Section 4.3). To identify informative features and find correlations, we further calculate information gain and gain ratio, correlate all features pairwise (see Section 5.1) and investigate the empirical cumulative distribution functions (ECDF) (see Section 5.2). We use the existing Adblock Plus blacklists for labeling in this step as the aim is to get an overview of the traffic properties of *all* services in our dataset.

Development of a classifier

After the initial feature analysis conducted in step 2, we focus on building a classifier. As we do not know a priori of which quality Adblock Plus' blacklists are, we use randomly selected, manually labeled Web services for developing the classifier.

3. To identify a suitable machine learning approach, we compare the classification performance of different machine learning (ML) methods using leave-one-out cross-validation (see Section 6.1).
4. Next, we identify the most suitable feature subset for the selected classifier by an exhaustive parameter search and measure the performance of the classifier (see Section 6.2).

Applying the classifier

We manually labeled services for the development of the classifier. Still, for use in practice, it would be preferable to avoid manual labeling. We show that this is indeed possible in the last two steps.

5. We train the classifier using the existing Adblock Plus blacklists as reference.
6. Finally, we compare the classification of our classifier against Adblock Plus' blacklists and discuss the results (see Section 7).

3.3 HTTP traffic features

Based on observations during preliminary work [19], we know that privacy-intrusive Web services are usually accessed by many users across several domains, and therefore include a higher diversity of client IP addresses and

Table 1. HTTP traffic features for this study. All features are aggregated per Web service. Note that basic counts are not normalized in this work for presentation purposes.

A: Basic counts		
1	number of HTTP requests	#requests
2	sent bytes (HTTP request bytes)	#sentBytes
3	received bytes (HTTP response bytes)	#recBytes
4	number of client IP addresses	#clients
5	domains in HTTP Referer	#referers
B: Average per request to service		
1	sent bytes per request	#sentBytes/req
2	received bytes per request	#recBytes/req
3	Referers per request	#referers/req
C: Average per client accessing service		
1	requests per client	#requests/client
2	sent bytes per client	#sentBytes/client
3	received bytes per client	#recBytes/client
4	Referers per client	#referers/client
D: Compound features		
1	% 3rd party requests	%3rdPartyReq
2	% 3rd party sent bytes	%3rdPartyBytes
3	% requests containing Cookie(s)	%cookies
4	received/sent bytes ratio	#rec/SentBytes

Referer domains than requests to regular Web services. Furthermore, we have noticed that especially analytics services barely deliver large responses. Hence, we expect that privacy-intrusive services have small response sizes as compared to regular services. With these observations in mind, we develop statistical features for Web service classification (see Table 1). The features are aggregated per Web service. We count the number of HTTP requests to the service, the sent/received bytes from/to the service, the IP addresses (clients) accessing the service, and the number of Referers (A.1-5). Furthermore, we calculate the average sent/received bytes and the #referers per HTTP request issued to the service (B.1-3) and per client accessing the service (C.2-4). We also calculate the average number of HTTP requests per client (C.1). Finally, we take a look at the ratio of the number of third-party requests to the total number of requests going to the service (D.1). We compute a similar ratio for bytes in third-party request to the total number of sent bytes (D.2). We determine the ratio of requests to the service that have at least one Cookie over the total number of requests (D.3) and the ratio of received to sent bytes (D.4).

Please note that to apply our model to a different dataset, the basic counts (A.1-5) need to be normalized by their corresponding total. As this results in tiny fractions that are difficult to interpret, we do not normalize the counts in this work for presentation reasons.

Since we introduced several variations of features measuring similar properties, we expect that an automatic classifier only requires a subset of the proposed features. However, we examine all 16 features in order to understand which specific features are most significant for classifying privacy-intrusive Web services.

3.4 Granularity of analysis

Our approach relies on traffic statistics of Web services. There are different possible levels of granularity for collecting such statistics.

Regarding the granularity at which we identify Web services, the first option is to collect statistics at the granularity of hosts, as they are contained in the headers of HTTP requests. However, this will not work well for Web services using a large number of subdomains for serving content, as for example *youtube.com*. The second option is to identify such services and aggregate subdomains depending on the number of requests being sent to them or the number of clients visiting them. But this would result in different levels of granularity. In contrast, we aim for an approach that is easily comprehensible. Therefore, we collect statistics at the granularity of second-level domains (SLD). For example, we aggregate requests to the third-level domains *www.example.com* and *static.example.com* to the SLD *example.com* and only collect statistics for *example.com*. There are top level domains that are subdivided, e.g., *.uk* or *.au*. We identify corresponding domains using the ICANN section of the public suffix list (http://publicsuffix.org/) and use the third-level domain as the "effective SLD". One could further aggregate SLDs, e.g., if they are owned by the same company. But we prefer to avoid further aggregation as this would make it difficult to compare to existing blacklists, which often operate at the granularity of "effective SLD" or even more fine-grained levels.

Regarding temporal granularity, we operate for this work at the granularity of days, meaning that we analyze our 24 h dataset as a whole, but our methodology is applicable to different time spans too.

Table 2. Overview of the 24-h trace collected in August 2013 on a campus network.

	up	down	#req.	#dom.	#clients
TRACE'13	75 GB	3.4 TB	60 M	103 k	15 k

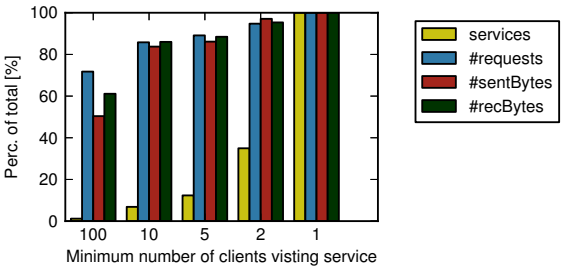


Fig. 3. Subsets of TRACE'13. The 1.2 k most popular services represented by SERVICES $\geq$ 100 CLIENTS account for more than 70 % of all requests. SERVICES $\geq$ 5 CLIENTS contains 13 k services accounting for almost 90 % of all requests and transmitted bytes. 65 % of all services are only visited by a single client.

4 Dataset

Next, we introduce the large packet traces that we use for evaluation in Section 4.1, discuss how we protect user privacy during our study in Section 4.2, and present the reference blacklist Adblock Plus and our manual classification methodology in Section 4.3.

4.1 Experimental data

4.1.1 Overview

We use 24 h of HTTP traffic recorded at the upstream router of a university campus for our analysis. The trace was created in August 2013, we refer to is as TRACE'13. The network is used by faculty staff of the university and students. We only analyze HTTP connections[2] initiated by IP addresses (clients) within the campus network to Web servers in the Internet, i.e., the HTTP activity of the clients in the university network. Baseline data on TRACE'13, which covers the HTTP activity of 15 k clients, is shown in Table 2. About 4 k clients solely issued requests with user agents unrelated to interactive browsing. We assume that these were mainly idle workstations (e.g., downloading software updates) or servers.

The packet traces were recorded in tcpdump/libpcap format [52]. To extract the HTTP traffic statistics, we use the BRO IDS [38] with a custom policy.

4.1.2 Subsets

For the evaluation of our approach, we compile two subsets of TRACE'13: SERVICES $\geq$ 100 CLIENTS and SERVICES $\geq$ 5 CLIENTS. The subscript indicates the minimum number of clients visiting the Web services in the corresponding dataset. For instance, SERVICES $\geq$ 5 CLIENTS contains all Web services that have been visited by at least 5 client IP addresses in the analyzed university network. Figure 3 shows statistics for these subsets.

We use two subsets to analyze if the classification model significantly changes when less popular services are incorporated. We focus for our analysis on the 13 k services visited by at least five clients, mainly for two reasons: *(i)* We do not expect our HTTP traffic statistics to be significant for Web services that are only visited by few users. *(ii)* We need to manually label some services to establish ground truth, which inevitably means that we see the domains accessed by users in our HTTP traces. Only inspecting services visited by at least 5 clients helps to protect the privacy of users in the analyzed network (see next section).

4.2 Protecting user privacy during analysis

Respecting the privacy of users when working with traces of real network traffic is of fundamental importance. We conduct our experiments according to a code of ethics protecting the privacy of users. In particular, we perform our analysis on an isolated system, never manually inspect payload or visited URLs of individual client IP addresses, anonymize client IP addresses prior to analysis, and only collect statistics per Web service and not per client, i.e., we refrain from building any client profiles. Still, even the SLD of a visited Web service could be considered privacy-sensitive, e.g., when a user accesses a private Web server from within the analyzed network. To mitigate this possibility, we only manually classify Web services accessed by at least 5 client IP addresses while establishing ground truth for our evaluation. Therefore, we do not become aware of Web services that could be very specific for one user in the analyzed network unless the user employs five

2 We focus on TCP port 80 HTTP connections as most HTTP traffic uses this port [31].

Table 3. Number of visited services listed by Adblock Plus.

	ABP_{EXACT}	$ABP_{PARTIAL}$
SERVICES $\geq$ 100 CLIENTS	354	643
SERVICES $\geq$ 5 CLIENTS	1171	2667

or more different IP address while accessing the private service. We think that five is a reasonable trade-off between user privacy and restrictions for analysis as this threshold allows us to conduct our analysis on a dataset containing the 13 k most popular Web services (SERVICES $\geq$ 5 CLIENTS).

4.3 Web service labeling

We rely on two data sources for Web service labeling, Adblock Plus blacklists and manual labeling, as we will discuss in this section.

4.3.1 Adblock Plus blacklists

We employ the blacklists coming with the popular anti-advertisement and analytics tool Adblock Plus [41] from August 2013. Specifically, we use the Adblock Plus filters EasyList (advertisement) and EasyPrivacy (analytics) as well as a specific EasyList for our region, all available on https://easylist.adblockplus.org/en/. To the best of our knowledge, these blacklists are manually maintained by a community of users. We use Adblock Plus blacklists, because Adblock Plus shows good blocking performance according to comparative benchmarks [21] and it is the most popular Firefox plug-in with 20 M users as of February 2015[3].

Adblock Plus contains rules for blocking and hiding elements. Hiding does in general not prevent a request from being issued, therefore we focus on blocking rules. Adblock Plus defines blocking rules on second-level domain, third-level domain, URL and path granularity, while we identify services on second-level domain granularity. This raises the question, how to map domains of different granularity for Web service labeling during learning and performance evaluation.

For comparing to Adblock Plus, we distinguish two cases:

1. **ABP_{exact}**: The entire second-level domain is blocked by Adblock Plus.
2. **$ABP_{partial}$**: The SLD appears in a blocking rule but the rule can be more specific, i.e., only block a subdomain or an URL containing the SLD.

In other words, if Adblock Plus lists the Web service *example.org/ad*, then the service *example.org* is in $ABP_{PARTIAL}$, but not in ABP_{EXACT}. Table 3 shows how many Web services are labeled as privacy-intrusive according to ABP_{EXACT} and $ABP_{PARTIAL}$.

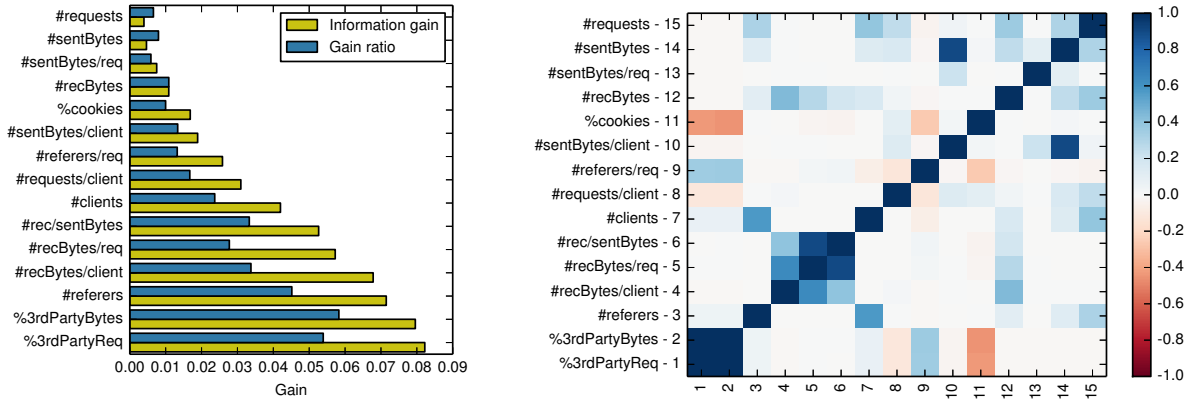
4.3.2 Manual labeling

As we do not know how complete Adblock Plus' blacklists are, we also conduct manual labeling on random samples of the Web services. We label services by visiting their company Web sites with a Web browser. If the corresponding Web site offers tracking or advertisement services, we label the service as *advertisement and analytics*, otherwise we label the service as *other*.

For most services, it is straightforward to find the company's Web site as there are two main cases: *(i)* the second level domain (SLD) hosts the company Web site (e.g., the SLD *criteo.com* is used for advertisement and analytics and the company Web site is reachable at *www.criteo.com*); *(ii)* companies that use a dedicated SLD for advertisement and analytics often forward to their Web site when the root directory of their advertisement and analytics SLD is visited (e.g., visiting *chartbeat.net* forwards to *chartbeat.com*, which is the company Web site) or they include a link to their company Web site (e.g., *quantserve.com* contains a link to the company Web site *quantcast.com*).

However, there are services for which finding the company Web site is more difficult, e.g., because the root directory of a SLD is just a blank page, visiting the SLD results in a "404 - Not Found", or no TCP connection can be established at all. In such and similar cases, we check DNS registration data, and search for the domain and terms listed on the loaded Web page to find the company Web site. For example, establishing an HTTP as well as HTTPS connection to *revsci.net* results in a timeout. But checking DNS registration data using `whois` shows that the admin email contact is associated with the domain *audiencescience.com*. A Web search for "revsci.net audiencescience.com" further leads to a page on *www.audiencescience.com* that explains how to embed a tag from *revsci.net*. Thus, we conclude that *www.audiencescience.com* is the company

3 https://addons.mozilla.org/en-US/firefox/extensions/?sort=users



(a) Information gain and gain ratio. **(b)** Pearson correlation.

Fig. 4. Feature evaluation for SERVICES $\geq$ 5 CLIENTS with respect to target class $\text{ABP}_{\text{EXACT}}$.

Web site corresponding to the SLD *revsci.net*. To be conservative, we treat a service as false positive during our evaluation in Section 7.1 if we remain uncertain about its class. We manually labeled several hundred Web services.

5 Properties of blacklisted Web services

As a first evaluation step, we statistically analyze properties of blacklisted Web services in this section. To get an intuition how well the features introduced in Section 3.3 perform for identifying advertisement and analytics services, we calculate information gain and Pearson correlation in Section 5.1 and analyze feature distributions in Section 5.2. We use the classification provided by the Adblock Plus blacklists in this section, as this allows us to analyze the properties of much more services than when labeling services manually.

5.1 Information gain and Pearson correlation

In order to calculate the information gain and the gain ratio [47], we first apply an entropy-based discretization [15]. Entropy-based discretization takes continuous features together with the classification as input and splits the domain of continuous features to minimize the classification entropy. As a result, highly informative continuous features are split into multiple intervals, while non-informative features are not split at all, resulting in a single discrete level. This has the advantage that continuous features are not only discretized but features with low value for classification are additionally removed from the initial feature space [11].

Figure 4a shows the features remaining after applying an entropy-based discretization with $\text{ABP}_{\text{EXACT}}$ as target class on the y-axis. The feature *#referers/client* has been identified as non-informative and is therefore not listed in the figure. The x-axis shows the information gain and the gain ratio with respect to $\text{ABP}_{\text{EXACT}}$ for the 15 remaining features. The corresponding Pearson correlation is shown in Figure 4b. The Pearson correlation coefficient r measures the linear correlation between two variables. The coefficient r takes values between -1 (perfect negative correlation) and $+1$ (perfect positive correlation). A value of $r = 0$ stands for no correlation.

We see in Figure 4 that *%3rdPartyReq* and *%3rdPartyBytes* are the most informative features. Both of these features target third-party requests and these features are nearly perfectly correlated. The feature *#referers* is the third-most informative feature. This feature is correlated with the features *#clients* (r=0.58) and *#requests* (r=0.32). The features *#recBytes/client*, *#recBytes/req* and *#rec/sentBytes* are all related to the amount of data a Web service transmits in responses to clients. This subset of features is again correlated (r=0.40, r=0.64, r=0.90) as shown in rows 4-6 of Figure 4b. Basic byte and request counts only provide little information with respect to $\text{ABP}_{\text{EXACT}}$, as we can see in Figure 4a.

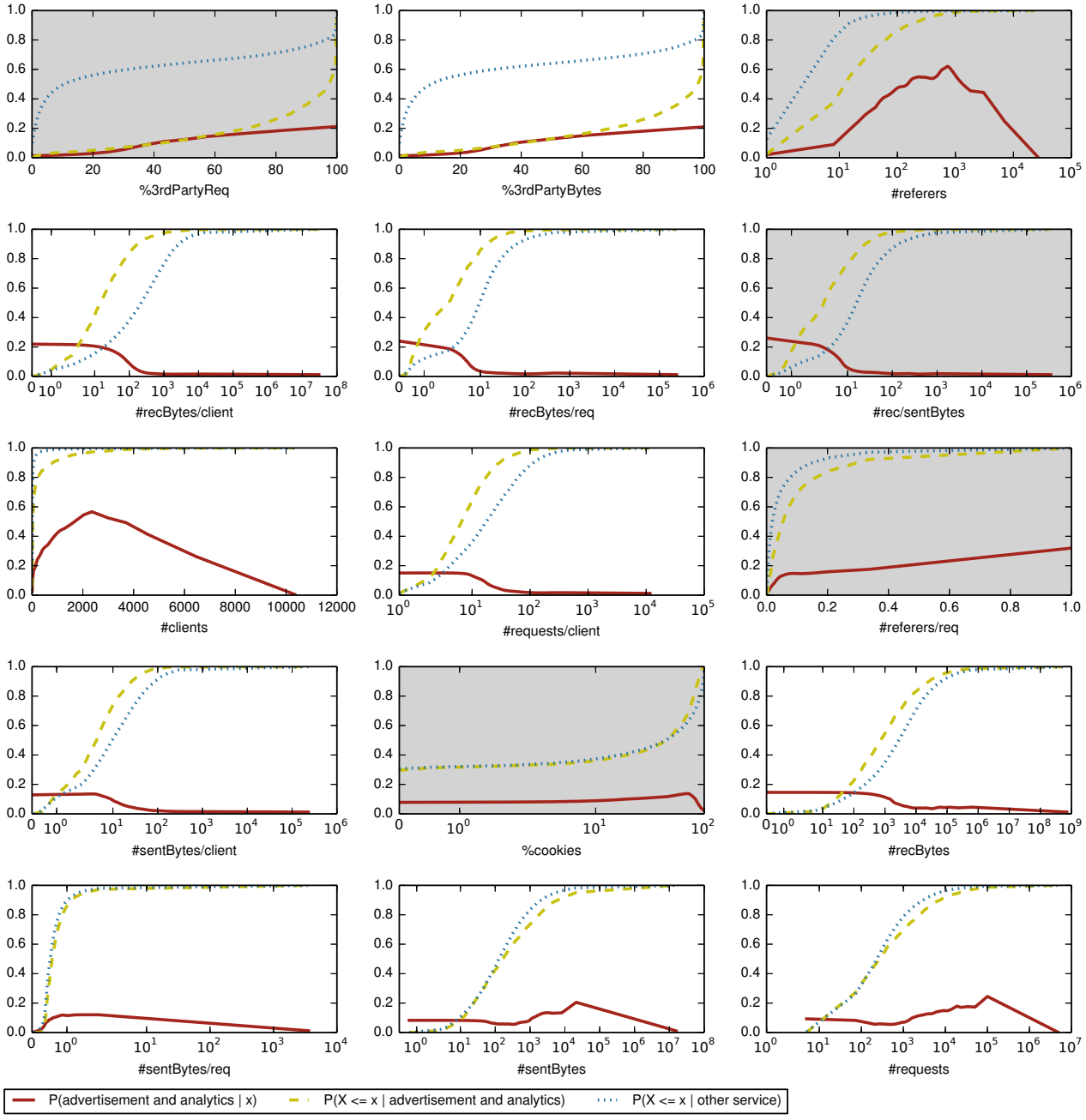


Fig. 5. Feature distributions for $\text{SERVICES}_{\geq 5\ \text{CLIENTS}}$. The solid red lines show the conditional probabilities for a service being an advertisement and analytics service according to $\text{ABP}_{\text{EXACT}}$ given certain feature values x. The dashed green and dotted blue lines show the empirical cumulative distribution functions (ECDF) for *advertisement and analytics* services and *other services* according to $\text{ABP}_{\text{EXACT}}$, respectively. Other services are all services that are not advertisement and analytics. 9 % of the services are in $\text{ABP}_{\text{EXACT}}$, i.e., the prior probability is $P(C = AA) = 0.09$. As a rule of thumb, the larger the area between the two ECDFs, the more discriminating power has the feature. Note that some features have a log-scale which deforms the area. Bytes are measured in KB. Features selected for our model (see Section 6.2) are marked by a gray background.

5.2 Distributions

In a second step, we discuss typical feature values for advertisement and analytics (AA) services, again labeled according to $\text{ABP}_{\text{EXACT}}$. Figure 5 shows *(i)* the empirical cumulative distribution functions (ECDF) separately for AA and all *other* services and *(ii)* the conditional probabilities that a service is an AA service given certain feature values. The ECDF $P(X \leq x)$ evaluated at

a value x tells the ratio of services having a feature value smaller or equal to x. For example, evaluating %3rdPartyReq in Figure 5 for $x = 10\,\%$ tells that about 50 % of all *other services* receive at most 10 % third-party requests, while the vast majority of AA services receive more than 10 % third-party requests.

$P(C = AA|X = x)$ is the conditional probability that a service is an AA service given a certain feature value x, as calculated by a naive Bayes classifier from the feature distributions. As a rule of thumb, a naive Bayes classifier will classify a Web service as advertisement and analytics if the service's feature values are rather in regions with a high conditional probability. That is, if the feature values x_i, where $i \in \{1, \ldots, 15\}$ indexes the features, are in regions with a high $P(C = AA|X_i = x_i)$.

The features *%3rdPartyReq* and *%3rdPartyBytes* show similar feature distributions. Third-party requests and bytes account for a high percentage of requests going to advertisement and analytics services, while about half of all other services receive less than 10 % third-party requests and bytes. This explains why we found in Figure 4a that these features are highly informative.

The feature *#referers* shows that there are more different Referers in HTTP requests to advertisement and analytics services than to most other services. That is, advertisement and analytics services are embedded (or linked to) by more services than most other services. This result is in-line with previous work [48] that investigated the most prominent trackers and showed that they are embedded on a large number of popular Web sites. As the red solid line indicates, especially in the region between 10^2 and 10^3 Referers, there are many more advertisement and analytics services than other services. Similarly, the feature *#clients* shows a trend that advertisement and analytics services receive requests from more clients than other services.

The features *#recBytes/client*, *#recBytes/req* and *#rec/sentBytes* relate to the amount of bytes that a service transmits back to clients. We can see on all three plots that advertisement and analytics services usually provide fewer bytes to clients than other services. The probability for a service being advertisement and analytics is highest for small values of these features. When averaged per number of clients or sent bytes, this trend becomes well visible. Still, the distributions of these three features are quite similar as the Pearson correlation in Figure 4b already suggested.

It might at first seem counter-intuitive that the median advertisement and analytics service has smaller values for *#requests/client* and *#sentBytes/client* than the median other service, as it should be expected that advertisement and analytics services collect a lot of data. However, advertisement and analytics services collect their data from a large client base and tracking a user may require as little as one request per page load. Most other Web services are visited by only few clients and a client that is browsing on a first party can easily cause hundreds to thousands of HTTP requests to a first party.

There are few services that almost have a ratio of 1 for the feature *#referers/req*. A ratio of 1 would indicate that every request going to the service has a different Referer. Advertisement and analytics services show a trend towards a higher ratio than other services. Interestingly, the results for the feature *%cookies* show that around 30 % of all services barely receive requests with Cookies.

6 Constructing a classifier

In this section, we test several popular machine learning techniques to determine which classifier works best (Section 6.1) and we evaluate if a reduction of the feature set can improve the classification results (Section 6.2). We use subsets of manually labeled data for these steps to avoid any bias possibly caused by Adblock Plus' classification.

6.1 Selecting a machine learning technique

For selecting a suitable machine learning technique, we randomly sample 1 % of the services in SERVICES ≥ 5 CLIENTS, manually classify the selected Web services as described in Section 4.3.2, and perform leave-one-out cross-validation using the Orange data mining tool set [11]. The target class for our analysis is *advertisement and analytics services*, i.e., a *true positive* is a correctly identified service that offers advertisement or analytics functionalities.

We consider the popular machine learning techniques naive Bayes with locally weighted regression (LOESS), logistic regression, a support vector machine (SVM) with a radial basis function kernel, and a classification tree with information gain for attribute selection. For background information on these machine learning approaches, we refer the reader to Hastie et al. [20] and Bishop [7].

Table 4. Comparison of classifiers using leave-one-out cross-validation with the fifteen features analyzed in Section 5. The columns show classification accuracy (CA), area under the receiver operating characteristic curve (AUC), F1-measure (F1), precision (Prec.), and recall for naive Bayes, logistic regression, SVM, and classification tree as classifiers. The maximum values are highlighted.

(a) Random sample of 127 services from SERVICES $\geq$ 5 CLIENTS.

	CA	AUC	F1	Prec.	Recall
Naive Bayes	85%	91%	61%	50%	79%
Logistic regression	85%	80%	24%	50%	16%
SVM	84%	77%	9%	33%	5%
Classification tree	86%	79%	57%	52%	63%

(b) Random sample of 122 services from SERVICES $\geq$ 100 CLIENTS.

	CA	AUC	F1	Prec.	Recall
Naive Bayes	84%	92%	83%	78%	89%
Logistic regression	81%	90%	79%	77%	81%
SVM	83%	90%	82%	75%	91%
Classification tree	80%	77%	77%	74%	81%

The results of our comparison are shown in Table 4. naive Bayes and classification tree show the best results. For reference, we also show the classification performance on a random subset from SERVICES $\geq$ 100 CLIENTS. In this dataset, the performance of the evaluated machine learning techniques is quite similar, with naive Bayes and logistic regression showing the best results.

The good performance of naive Bayes compared to the more advanced methods is not obvious. A naive Bayes classifier builds upon the "naive" assumption that features are conditionally independent given the class, which clearly does not hold for the features we provided as input, as the analysis in Section 5.1 showed. Still, the good performance of naive Bayes compared to the more advanced approaches is in-line with prior work reporting that "it (Bayes) performs surprisingly well in many domains containing clear attribute dependences" [13]. There are multiple explanations for the high classification accuracy in the presence of correlated features. First, inaccurate probability estimations have no or limited impact as long as the maximum probability is assigned to the correct class; second, dependences between attributes cancel each other out if they are evenly distributed in the classes [13, 55].

For the remainder of our work we use naive Bayes because of *(i)* its good performance and *(ii)* the simplicity of its classification model, which allows us to analyze and discuss classification results in detail.

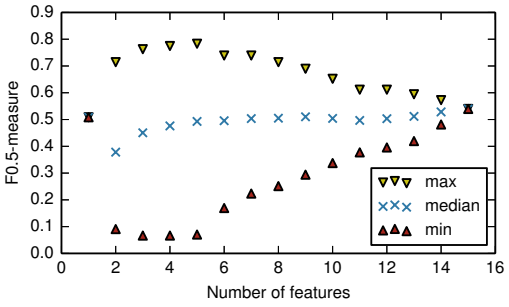


Fig. 6. $F_{0.5}$-measure for all feature subsets measured on a random sample of 127 services from SERVICES $\geq$ 5 CLIENTS. The x-axis shows the size of the subsets and the y-axis the maximum, median, and minimum $F_{0.5}$-measure. Details on the best subset consisting of five features are listed in Table 5.

6.2 Feature reduction

So far, we used all features that the entropy-based discretization in Section 5.1 has identified as being informative. But some of the features are highly correlated while others have a small information gain and gain ratio as can be seen in Figure 4. Including such features in the classification model increases the risk for overfitting the classifier and also makes the model more complex to understand. We measure in this step the classification performance for all possible feature subsets to empirically determine the best feature set.

We use the $F_{0.5}$-measure to evaluate classification performance. A F_β-measure combines the precision and recall of a classifier to measure its classification performance. The $F_{0.5}$-measure is the typical value used to weight precision higher than recall. In other words, we are aiming towards a low false positive rate when identifying privacy-intrusive services. We perform an exhaustive search by building all 33 k subsets of the 15 features and conducting leave-one-out cross- validation with naive Bayes on a random sample of 127 services from SERVICES $\geq$ 5 CLIENTS. In this sample, 9 services are in $\text{ABP}_{\text{EXACT}}$ and 29 in $\text{ABP}_{\text{PARTIAL}}$.

Figure 6 shows the results of the parameter search. Subsets not identifying any privacy-intrusive services are not shown in the Figure, since they have an undefined $F_{0.5}$-measure. We see that using five features results in the highest $F_{0.5}$-measure. The detailed performance of the corresponding features is shown in Table 5. We want to point out that the five selected fea-

Table 5. Performance of the best feature set measured using leave-one-out cross-validation against manually labeled data.

Selected features
%3rdPartyReq, #referers, #rec/sentBytes, #referers/req, %cookies

Classification performance			
Dataset	F0.5	Precision	Recall
1 % of SERVICES $\geq$ 5 CLIENTS (feature selection)	0.78	0.81	0.68
10 % of SERVICES $\geq$ 100 CLIENTS (evaluation)	0.84	0.83	0.85

Table 6. Complementing Adblock Plus' blacklists.

(a) Comparison to Adblock Plus. Services that are identified by our classifier are shown in the left columns, services that are not identified by our classifier but that are listed by Adblock Plus are shown in the two rightmost columns.

		$\in$ Identified			$\notin$ Identified	
Dataset	Total identified	$\notin \mathrm{ABP}_{\mathrm{EXACT}}$	$\notin \mathrm{ABP}_{\mathrm{PARTIAL}}$	$\notin \mathrm{ABP}_{\mathrm{PARTIAL}} \wedge \in \mathrm{ABP}^{2015}_{\mathrm{PARTIAL}}$	$\in \mathrm{ABP}_{\mathrm{EXACT}}$	$\in \mathrm{ABP}_{\mathrm{PARTIAL}}$
SERVICES $\geq$ 100 CLIENTS	421	142	86	18	75	308
SERVICES $\geq$ 5 CLIENTS	1081	537	410	70	627	1996

(b) Manual inspection of newly identified services not in $\mathrm{ABP}_{\mathrm{PARTIAL}}$.

Dataset	Advertisement and analytics	Service utility	Online shopping and auctions	CDN and apps	User centric	Other
SERVICES $\geq$ 100 CLIENTS	52	11	8	5	3	7
SERVICES $\geq$ 5 CLIENTS (10 % rnd. sample)	19	2	4	1	2	13

tures, which we are going to use for the remainder of this work, are actually only slightly correlated as Figure 4b shows.

After selecting the features using a random sample from SERVICES $\geq$ 5 CLIENTS, we use a disjoint manually labeled random sample from SERVICES $\geq$ 100 CLIENTS to evaluate precision and recall. For this dataset, precision and recall are 83 % and 85 %, respectively, resulting in a $F_{0.5}$-measure of 84 %. This sample consists of 122 services, 38 services are in $\mathrm{ABP}_{\mathrm{EXACT}}$ and 64 in $\mathrm{ABP}_{\mathrm{PARTIAL}}$.

7 Complementing Adblock Plus' blacklists

In the previous section, we built our classifier using manually labeled data. But for use in practice, it would be preferable to avoid any need for manual labeling. Therefore, in Section 7.1, we evaluate how many new privacy-intrusive Web services our classifier can identify by learning from existing Adblock Plus blacklists, i.e., without manual labeling. In Section 7.2, we conclude the evaluation by discussing factors that influence the recall of our classifier.

7.1 Newly identified services

We train a naive Bayes classifier using the five features identified in Table 5 and Adblock Plus blacklists from August 2013, which corresponds to the up-to-date Adblock Plus blacklist at the time of recording the analyzed traffic. Then we apply our classifier to the data we just learned from. We point out that this does not correspond to traditional machine learning evaluation methodology. However, our aim is not to statistically evaluate the classifier but to find services that Adblock Plus misses without manually labeling a training set.

Our aim is to build a conservative classifier, therefore we use the Adblock Plus category $\mathrm{ABP}_{\mathrm{EXACT}}$ for training, which only contains exact domain matches (see Section 4.3.1). For the evaluation of our classifier, we later compare the identified services with $\mathrm{ABP}_{\mathrm{PARTIAL}}$. Hence, we only claim that our classifier has identified a new privacy-intrusive second level-domain, if no subdomain has already been blocked by Adblock Plus.

We conduct the described analysis separately for SERVICES $\geq$ 5 CLIENTS and SERVICES $\geq$ 100 CLIENTS. The results are listed in Table 6a. In SERVICES $\geq$ 100 CLIENTS, our detector identifies 421 privacy-intrusive services. 86 of these are not listed by $\mathrm{ABP}_{\mathrm{PARTIAL}}$. Manual verification of these services (see Table 6b) shows that indeed more than 60 % are distinct advertisement and analytics services. In SERVICES $\geq$ 5 CLIENTS, our detector identifies 1081 privacy-intrusive services, 410 of these are not listed by $\mathrm{ABP}_{\mathrm{PARTIAL}}$. We randomly sample 10 % of the newly identified services and classify them manu-

ally. We find that 46 % of the services in this random sample are indeed advertisement and analytics services not listed by $\text{ABP}_{\text{PARTIAL}}$. Therefore, we estimate that our approach identifies in the order of 200 new advertisement and analytics services.

At the same time, our classifier misses 627 services that were originally marked as advertisement and analytics services in the learning set. As the Web landscape is very diverse, it is not surprising that our automated classifier, which is only based on high-level HTTP traffic statistics, can not compete with a manually compiled blacklist in terms of recall.

We also compare our classification results to Adblock Plus blacklists as of February 2015. We find that 70 services that our approach would have identified in 2013 but that were not listed by $\text{ABP}_{\text{PARTIAL}}$ at that time, have in the meantime been added to Adblock Plus. Further, our classifier finds a number of Web site utility services as well as shopping and auction platforms. We analyze the corresponding services in $\text{SERVICES}_{\geq 100\ \text{CLIENTS}}$ and find that they differ especially regarding the feature *#referers/req* from the average Web service. Many of these services have a higher value for this feature than the average Web service. The reason why so many Web shopping sites match the conditions applied by our detector is that these Web services host advertisements or other elements related to their online shops and these objects are included by other Web sites. Also many Web site utility services such as *flattr.com* and user centric services such as *gravatar.com* behave in terms of our statistics similar to advertisement and analytics services. While Flattr explicitly state on their blog that they do not use the collected data to track users, we found no such statement for Gravatar and thus do not know whether they exploit their good vantage point for user tracking. Yet, these services also provide utility which may be useful to some users. Consequently, different users may have different opinions about the privacy trade-off of each of these services.

Note however that services offering Web site utility functionality do not always stand out so clearly: the popular online social networks (OSNs) Google, Facebook and Twitter all provide some kind of "like-button", which they also use to track users. Our classifier cannot detect these OSN trackers because their request mix is dominated by clients interacting directly with these services.

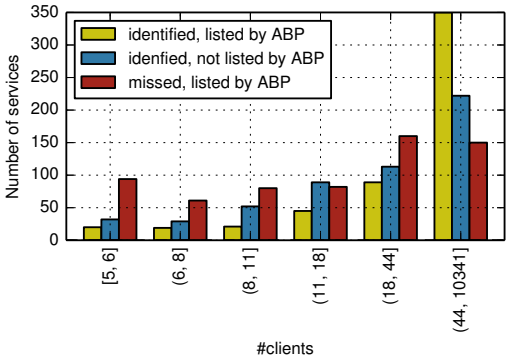


Fig. 7. Histogram of *#clients* accessing a service vs. number of advertisement and analytics services listed by $\text{ABP}_{\text{EXACT}}$ and identified by our classifier for dataset $\text{SERVICES}_{\geq 5\ \text{CLIENTS}}$. Every histogram bin relies on 2.1k services.

7.2 Impact of service popularity on recall

As our classifier misses a large number of services that were in the original learning set, we investigate recall with respect to the number of clients (*#clients*) accessing a service and the number of other services embedding a service's content (*#referers*).

Figure 7 shows the distribution for *#clients*. The figure shows that most listed Adblock Plus services can be found among the Web services accessed by many clients. At the same time, our classifier also identifies most privacy-intrusive services among the popular Web services. The distribution for *#referers* is very similar. We conclude that our methodology will mainly identify privacy-intrusive services accessed by many clients and being present on many Web sites, while unpopular privacy-intrusive services are less likely to be identified.

8 Use cases

We envision two main use cases for the application of our method: global and local use.

8.1 Global use

As our method solely relies on high-level Web service statistics, having a centralized repository of traffic statistics for every Web service, aggregated from contributions of volunteers, is sufficient to perform both the machine learning phase as well as the subsequent application of the classifier on individual Web services.

For example, the Mozilla Lightbeam project aims at building exactly such a centralized repository with the help of a browser plug-in [34]. With its current data format (data_format.v1.2.md[4]), the Lightbeam plug-in collects all relevant features except for byte counts, but it does no longer transmit data to the Lightbeam database as of January 2015. Further, also anti-virus software vendors that collect statistics on Web sites visited by their users could collect corresponding service features.

In order to find new candidates for blacklists, a blacklist maintainer could *(i)* regularly download aggregated Web service statistics from a corresponding crowd-sourced database, *(ii)* train a classifier by using the downloaded statistics and an existing blacklist as reference[5], and *(iii)* apply the classification model to the downloaded statistics to find new candidates.

Required statistics are calculated *per Web service* and *per Web service and user* (to count the number of users visiting a Web service), but no statistics need to be calculated *per individual user*. This has the advantage that there is no need to store the mapping between a global user identifier and the visited services. That is, a browser plug-in that submits statistics to a corresponding crowd-sourced database can use a different user identifier for every Web service and the information that a crowd-sourced database needs to provide to calculate our features is not sufficient to build user browsing profiles.

8.2 Local use

As second use case, organizations can collect traffic statistics from their own proxy servers or egress gateways. Then, they can either build a classifier by learning the required feature distributions from their own traffic or by using feature distributions collected in a different network, as the ones we present. All that is required to train a naive Bayes classifier are the distributions and the prior probability as presented in Figure 5. The benefit of applying our technique locally is that an organization can identify privacy-intrusive services that are specific to their traffic and not listed on general blacklists.

4 https://github.com/mozilla/lightbeam/blob/master/doc/data_format.v1.2.md, Last updated: 2014-05-08

5 In order to train a classifier, statistics about traffic to blacklisted and other Web services need to be available. If all known blacklisted services have been blocked and statistics on blacklisted services are no longer available, one can alternatively use probability distributions recorded earlier or in a different network, as the ones we present in Figure 5.

9 Related Work

Mayer and Mitchell provide a survey [32] of policies, technologies, and risks involved in Web tracking. In addition, we discuss selected works related to ours in the rest of this section.

HTTP traffic characterization

The characterization of HTTP traffic, especially with respect to accessed Web sites, downloaded objects, and implications on caching is a long-lived and well-explored topic [4, 5, 8, 10, 14, 22, 31]. In contrast, we aim at identifying privacy-intrusive Web services.

Traffic classification using machine learning

Applying machine learning for classifying network traffic is not new. Karagiannis et al. [23] analyze traffic patterns at the social, functional, and application level allowing them to classify 80-90 % of the traffic with more than 95 % accuracy. However, while one of their categories is "Web", they do not distinguish different types of Web services. Salgarelli et al. [49] discuss how to reproduce traffic classification results and how to evaluate classifier performance. Bernaille et al. [6] apply machine learning to classify traffic only based on the first five packets of a TCP connection. Among other categories, they distinguish between HTTP and HTTPS, but they do no further classify Web services. Kim et al. [24] compare several classification methods and show how different approaches can be combined to achieve better performance. In contrast to these works, we focus on Web traffic and apply machine learning to identify unlisted advertisement and analytics services.

Tracking and privacy

Web tracking and privacy received considerable attention in the scientific community over the past years, covering general problems related to tracking [25], characterizing the information leakage to third parties [26, 27], characterizing the third parties themselves [48], and corresponding mitigation strategies [26, 27, 48]. All of these approaches are based on active measurement towards a

limited number of top Web sites, focusing on the largest Web trackers. In contrast, we utilize machine learning to automatically pin-point a large amount of Web trackers, thus complementing existing blacklists and leading to a better understanding of the overall tracker landscape.

Tran et al. [54] use JavaScript tainting to find trackers of privacy-relevant information, and can pin-point one tracking Web site in addition to the services listed by Ghostery. In contrast, our machine learning approach reveals more than 400 new sites showing tracker-like behavior, and based on a random sample we estimate that 46 % are indeed traditional advertisement and analytics services.

The advertisement ecosystem

Recently, researchers started investigating the economics behind Web advertisement, including modeling the user's value to advertisement companies [18] and extracting pricing information from the real-time bidding process for advertisement placement [36].

Evading trackers

Different solutions exist to evade trackers, including blocking of JavaScript execution [43], blocking the access to certain URLs or domains [12, 41, 42, 46], sharing Internet-facing IP addresses [12, 53], or suppressing referer headers using a browser plug-in[6]. Not all approaches are sufficient on their own. Our machine learning technique aims at complementing the blacklists used by various tools that selectively block Web requests.

Predictive blacklisting

There is a large body of related work on predictive blacklisting. Predictive blacklisting strategies intend to forecast attack sources, such as URLs for phishing attacks [44]. Zhang et al. [56] present an approach called highly predictive blacklisting, which applies a page rank-style algorithm to identify malicious IP addresses. Their approach has been integrated in DShield, a popular community-based firewall log correlation system. Ma et al. [29] propose a machine learning approach to detect malicious Web sites from suspicious URLs. The classification of the Web sites is based on host-based and lexical features, such as WHOIS, domain name, IP address and geographic properties. They achieve a classification accuracy of 95-99%.

Pao et al. [37] distinguish between malicious and benign Web sites by estimating the conditional Kolmogorov complexity of URL strings. Ma et al. [30] generate personalized, predictive blacklists for individual networks by correlating previous attacks captured in a honeynet. Finally, Soldo et al. [51] propose a blacklisting recommendation system which is based on a time-series model that accounts for temporal dynamics and two neighborhood-based models.

Unlike related work, we focus on privacy-intrusive Web services such as advertisement services and trackers which do not perform severe attacks. Hence, we do not necessarily need to prevent any access to such services, but can allow for some accesses in order to investigate the HTTP traffic to/from the investigated service. Therefore, our methodology fundamentally differs from related work on predictive blacklisting.

Other work

The Mozilla Lightbeam browser plug-in [34] collects connection data and visualizes third parties in an interactive graph. Lightbeam does not support classification of Web sites yet. However, our approach could work on the collected data to identify trackers (see Section 8).

10 Conclusion

In this work, we investigate which statistical HTTP traffic features and classifiers are suitable to identify privacy-intrusive services. We find a naive Bayes classifier with a set of five features to work best, achieving a precision and recall of up to 83 % and 85 %, respectively. By training our classifier on existing Adblock Plus blacklists, we find more than 400 new possibly privacy-intrusive services. Manual verification shows that around 200 of the suggested services are indeed traditional advertisement and analytics services. Among the other services, we mainly find shopping sites providing advertisement content, as well as Web site and user utility services. The Web site and user utility services often only provide marginal functionalities such as a global avatar. It is an open question if these services should be classified as privacy-intrusive.

6 For example, see https://addons.mozilla.org/en-US/firefox/search/?q=referer (accessed: 2015-05-14) for Mozilla Firefox and https://chrome.google.com/webstore/search/referer?_category=extensions (accessed: 2015-05-14) for Chrome .

We evaluate our technique using traffic traces captured on a central gateway of a large campus network. Organizations can directly profit from our results by implementing our approach in a similar way. Our approach can also work with data collected in a distributed fashion, hence, we believe that also individual users can profit from our results.

Acknowledgement

This work was partially supported by the Zurich Information Security Center. It represents the views of the authors.

References

[1] J. Abbatiello. RefControl – Firefox Add-on. https://addons.mozilla.org/de/firefox/addon/refcontrol. Accessed: 2015-02-14.

[2] G. Acar, C. Eubank, S. Englehardt, M. Juarez, A. Narayanan, and C. Diaz. The web never forgets: Persistent tracking mechanisms in the wild. In *Proc. CCS '14*, pages 674–689, 2014.

[3] L. Andrews. Facebook Is Using You. New York Times (2012-02-04), http://www.nytimes.com/2012/02/05/opinion/sunday/facebook-is-using-you.html. Accessed: 2015-02-14.

[4] M. F. Arlitt and C. L. Williamson. Web server workload characterization: the search for invariants. In *Proc. SIGMETRICS '96*, pages 126–137, 1996.

[5] P. Barford, A. Bestavros, A. Bradley, and M. Crovella. Changes in web client access patterns: Characteristics and caching implications. *World Wide Web*, 2(1-2):15–28, 1999.

[6] L. Bernaille, R. Teixeira, I. Akodkenou, A. Soule, and K. Salamatian. Traffic classification on the fly. *SIGCOMM Comput. Commun. Rev.*, 36(2):23–26, Apr. 2006.

[7] C. M. Bishop. *Pattern recognition and machine learning*. Springer, 2006.

[8] M. Butkiewicz, H. V. Madhyastha, and V. Sekar. Understanding website complexity: Measurements, metrics, and implications. In *Proc. IMC '11*, pages 313–328, 2011.

[9] R. Cookson. Google, Microsoft and Amazon pay to get around ad blocking tool. Financial Times (2015-02-01), http://www.ft.com/cms/s/0/80a8ce54-a61d-11e4-9bd3-00144feab7de.html. Accessed: 2015-02-15.

[10] M. E. Crovella and A. Bestavros. Self-similarity in world wide web traffic: evidence and possible causes. *IEEE/ACM Trans. Netw.*, 5(6):835–846, 1997.

[11] J. Demšar, T. Curk, A. Erjavec, Črt Gorup, T. Hočevar, M. Milutinovič, M. Možina, M. Polajnar, M. Toplak, A. Starič, M. Štajdohar, L. Umek, L. Žagar, J. Žbontar, M. Žitnik, and B. Zupan. Orange: Data mining toolbox in python. *Journal of Machine Learning Research*, 14:2349–2353, 2013.

[12] Disconnect | Online Privacy & Security. https://disconnect.me/.

[13] P. Domingos and M. Pazzani. On the optimality of the simple bayesian classifier under zero-one loss. *Machine learning*, 29(2-3):103–130, 1997.

[14] F. Douglis, A. Feldmann, B. Krishnamurthy, and J. Mogul. Rate of change and other metrics: a live study of the world wide web. In *Proc. USENIX Symp. on Internet Technologies and Systems*, Dec. 1997.

[15] U. M. Fayyad and K. B. Irani. Multi-interval discretization of continuous-valued attributes for classification learning. In *Proc. 13th Int. Joint Conf. on Artificial Intelligence*, pages 1022–1027, 1993.

[16] M. Fertik. The Rich See a Different Internet Than the Poor. Scientific American Volume 308, Issue 2, http://www.scientificamerican.com/article/rich-see-different-internet-than-the-poor/. Accessed: 2015-02-14.

[17] R. Fielding and J. Reschke. Hypertext Transfer Protocol (HTTP/1.1): Semantics and Content. RFC 7231.

[18] P. Gill, V. Erramilli, A. Chaintreau, B. Krishnamurthy, K. Papagiannaki, and P. Rodriguez. Follow the money: Understanding economics of online aggregation and advertising. In *Proc. IMC '13*, pages 141–148, 2013.

[19] D. Gugelmann, B. Ager, and V. Lenders. Towards classifying third-party web services at scale. In *Proc. CoNEXT Student Workshop '14*, pages 34–36, 2014.

[20] T. Hastie, R. Tibshirani, and J. Friedman. *The elements of statistical learning*, volume 2. Springer, 2009.

[21] R. Hill. Comparative benchmarks against widely used blockers: Top 15 Most Popular News Websites. https://github.com/gorhill/httpswitchboard/wiki/Comparative-benchmarks-against-widely-used-blockers:-Top-15-Most-Popular-News-Websites. Accessed: 2015-02-13.

[22] S. Ihm and V. S. Pai. Towards understanding modern web traffic. In *Proc. IMC '11*, pages 295–312, 2011.

[23] T. Karagiannis, K. Papagiannaki, and M. Faloutsos. Blinc: multilevel traffic classification in the dark. *SIGCOMM Comput. Commun. Rev.*, 35(4):229–240, 2005.

[24] H. Kim, K. Claffy, M. Fomenkov, D. Barman, M. Faloutsos, and K. Lee. Internet traffic classification demystified: Myths, caveats, and the best practices. In *Proc. ACM CoNEXT '08*, pages 11:1–11:12, 2008.

[25] B. Krishnamurthy. I know what you will do next summer. *SIGCOMM Comput. Commun. Rev.*, 40(5):65–70, 2010.

[26] B. Krishnamurthy, D. Malandrino, and C. E. Wills. Measuring privacy loss and the impact of privacy protection in web browsing. In *Proc. 3rd Symp. on Usable Privacy and Security (SOUPS '07)*, pages 52–63, 2007.

[27] B. Krishnamurthy, K. Naryshkin, and C. E. Wills. Privacy leakage vs. protection measures: the growing disconnect. In *Proc. Web 2.0 Security and Privacy Workshop*, 2011.

[28] T. Libert. Privacy implications of health information seeking on the web. *Commun. ACM*, 58(3):68–77, 2015.

[29] J. Ma, L. K. Saul, S. Savage, and G. M. Voelker. Beyond blacklists: Learning to detect malicious web sites from suspicious urls. In *Proc. 15th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*, KDD '09, pages 1245–1254, 2009.

[30] X. Ma, J. Zhu, Z. Wan, J. Tao, X. Guan, and Q. Zheng. Honeynet-based collaborative defense using improved highly predictive blacklisting algorithm. In *8th World Congr. on Intelligent Control and Automation*, WCICA '10, pages 1283–1288, 2010.

[31] G. Maier, A. Feldmann, V. Paxson, and M. Allman. On dominant characteristics of residential broadband internet traffic. In *Proc. IMC '09*, pages 90–102, 2009.

[32] J. R. Mayer and J. C. Mitchell. Third-party web tracking: Policy and technology. In *Proc. SP '12*, pages 413–427, 2012.

[33] J. Mikians, L. Gyarmati, V. Erramilli, and N. Laoutaris. Detecting price and search discrimination on the internet. In *Proc. HotNets-XI '12*, pages 79–84, 2012.

[34] Mozilla | Lightbeam for Firefox. https://www.mozilla.org/en-US/lightbeam/. Accessed: 2015-04-28.

[35] L. Olejnik, C. Castelluccia, and A. Janc. Why johnny can't browse in peace: On the uniqueness of web browsing history patterns. In *Proc. HotPETs '12*, 2012.

[36] L. Olejnik, T. Minh-Dung, and C. Castelluccia. Selling off privacy at auction. In *Proc. NDSS '14*, 2014.

[37] H.-K. Pao, Y.-L. Chou, and Y.-J. Lee. Malicious url detection based on kolmogorov complexity estimation. In *Proc. Int. Conf. on Web Intelligence and Intelligent Agent Technology*, WI-IAT '12, pages 380–387, 2012.

[38] V. Paxson. Bro: a system for detecting network intruders in real-time. *Computer Networks*, 31(23-24):2435–2463, 1999.

[39] D. Peck. They're Watching You at Work. The Atlantic (2013-11-20), http://www.theatlantic.com/magazine/archive/2013/12/theyre-watching-you-at-work/354681/. Accessed: 2015-02-14.

[40] Adblock. https://getadblock.com.

[41] Adblock Plus. https://adblockplus.org.

[42] Ghostery. https://www.ghostery.com.

[43] NoScript. https://noscript.net.

[44] P. Prakash, M. Kumar, R. R. Kompella, and M. Gupta. Phishnet: Predictive blacklisting to detect phishing attacks. In *Proc. INFOCOM '10*, pages 1–5, 2010.

[45] R. Pries, Z. Magyari, and P. Tran-Gia. An http web traffic model based on the top one million visited web pages. In *Proc. EURO-NGI Conf. Next Generation Internet (NGI)*, pages 133–139, 2012.

[46] Electronic Frontier Foundation | Privacy Badger. https://www.eff.org/de/node/73969. Accessed: 2015-02-13.

[47] J. R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers Inc., 1993.

[48] F. Roesner, T. Kohno, and D. Wetherall. Detecting and defending against third-party tracking on the web. In *Proc. NSDI '12*, 2012.

[49] L. Salgarelli, F. Gringoli, and T. Karagiannis. Comparing traffic classifiers. *SIGCOMM Comput. Commun. Rev.*, 37(3):65–68, 2007.

[50] L. Scism and M. Maremont. Insurers Test Data Profiles to Identify Risky Clients. Wall Street Journal (2010-11-19), http://www.wsj.com/articles/SB10001424052748704648604575620750998072986. Accessed: 2015-02-14.

[51] F. Soldo, A. Le, and A. Markopoulou. Blacklisting recommendation system: Using spatio-temporal patterns to predict future attacks. *J. on Selected Areas in Commun.*, 29(7):1423–1437, 2011.

[52] Tcpdump/Libpcap. http://www.tcpdump.org.

[53] Tor | Anonymity Online. https://www.torproject.org.

[54] M. Tran, X. Dong, Z. Liang, and X. Jiang. Tracking the trackers: Fast and scalable dynamic analysis of web content for privacy violations. In *Proc. Conf. on Applied Cryptography and Network Security*, ACNS '12, pages 418–435, 2012.

[55] H. Zhang. The Optimality of Naive Bayes. In *Proc. FLAIRS '04*, 2004.

[56] J. Zhang, P. A. Porras, and J. Ullrich. Highly predictive blacklisting. In *Proc. USENIX Security '08*, 2008.