Towards Understanding Upstream Web Traffic

David Gugelmann*, Bernhard Ager*, Vincent Lenders† and Markus Happe*
*ETH Zurich, Zurich, Switzerland, Email: {gugelmann, bager, mhappe}@tik.ee.ethz.ch
†armasuisse, Thun, Switzerland, Email: vincent.lenders@armasuisse.ch

***Abstract*—While downstream Web traffic has been studied in detail, upstream Web traffic has not received much attention yet. We argue that upstream traffic deserves the same or even higher attention since data flows towards Web servers generally entail privacy-relevant user information. Our aim is to understand where to and how much data users send to Web services. To this end, we examine HTTP(S) requests of two 24 hour traces recorded at a gateway of a campus network. As HTTP is highly repetitive, we introduce a scalable approach to remove redundant parts from upstream Web traffic, yielding an approximation of actual information flow. We identify thirteen classes of Web services covering up to 95 % of all outgoing HTTP information. Our methodology further allows to quantify and compare the share of information different Web service classes receive. We find that advertisement and analytics services receive two times more information during Web browsing than all first party Web services together.**

Keywords*—*Upstream Web traffic; Web service classification; HTTP traffic; HTTPS traffic; user tracking; network measurement

I. Introduction

In earlier times, the Web was relatively simple. Web pages were published by a savvy set of people and mainly consisted of text and images all hosted by the same server. However, the Web has changed. Web pages today consist of dozens to hundreds of embedded objects loaded from different dedicated services such as content distribution networks, video portals, and advertisement and analytics services. This results in a large footprint when loading Web pages, i.e., HTTP requests to many services. At the same time, regular users are no longer only in the role of consumers but increasingly provide contents and data. For example, by posting on blogs and online social networks, or by uploading to video and cloud storage services. In the previous Web model, the network traffic was highly asymmetric with most of the data flowing to the users. Perhaps not surprisingly, previous research efforts that aimed at characterizing HTTP traffic (e.g., [1]–[7]) were highly focused on downstream Web traffic. While downstream Web traffic is still more voluminous than upstream traffic today, we argue that the upstream Web traffic deserves also particular attention because upload traffic is becoming more and more prevalent and generally entails privacy-related user information. Understanding upstream Web traffic can tell us about the amount and type of personal information that users are disclosing over the Web while making use of it. To the best of our knowledge, we are the first to scrutinize the complete upstream Web traffic of a large network. We analyze two 24 h traces recorded in a university campus.

Our aim is to understand the classes of Web services to which users transmit information. Our contributions are:

1) We develop a scalable method to reduce redundancy in HTTP requests to approximate the actual amount of information transmitted to Web servers (see Section V-A). Our heuristic reduces the overall byte volume in HTTP request headers by more than a factor of 17, which makes Web services receiving actual non-redundant data stand out much more clearly (see Section V-B). Applying our heuristic especially impacts the ranking of popular news Web sites (see Section V-C).
2) We find that upstream Web traffic is dominated by very few clients and services (similarly to downstream traffic [8]): Only ten client-server pairs account for 50 - 80 % of all non-redundant upstream HTTP traffic and for around 40 % of the upstream HTTPS traffic. *Dropbox* and *Google* receive almost 80 % of all HTTPS upstream traffic (see Section V-D).
3) We classify Web services and analyze data in HTTP request headers and bodies separately for first- and third-party Web services. This allows us to find traffic patterns that are otherwise not visible. For example, we find an anti-malware software that receives a large amount of data on visited URLs (see Section V-E).
4) We analyze the ecosystem of contacted third-party services when loading Web pages and find that advertisement and analytics services receive about 60 % of all third-party information and two times more information than all first-party services together (see Section V-F).

II. Information flow to Web services

The focus of this paper is the characterization of Web traffic that is flowing out of users' machines at a campus network to Web services in the Internet. During Web browsing, the users either trigger HTTP and HTTPS requests to these services directly, e.g., by clicking on a link, or the Web browser automatically issues requests to load embedded elements. Moreover it is possible that software on user machines automatically generates traffic, e.g., for checking and downloading software updates. We do not differentiate between user- and software-initiated traffic as both sorts of traffic may contain user-related information.

When estimating the amount of information transmitted to a Web service, the raw network data volume is clearly an indicator. However, it is well known that HTTP Web traffic is highly redundant[1] [5], [10]. Therefore, we introduce a heuristic to estimate *information flow* in a HTTP request to a Web

[1]Note that header compression for HTTP/2 [9] reduces redundancy only in some retransmitted bytestrings, therefore we expect considerable redundancy also with the upcoming HTTP/2 standard (see Section V-B).

978-1-4799-5344-8/15/$31.00 ©2015 IEEE

service X as the total HTTP request size minus the size of all retransmitted and reflected bytestrings; retransmitted bytestrings are strings previously observed in requests *to* Web service X, reflected bytestrings are strings previously observed in responses *from* Web service X (see Section V-A for details). We refer to retransmitted and reflected bytestrings as *redundant* parts.

Our work strives to answer the following questions:

1) Which classes of Web services receive most information?
2) What share of outgoing information caused by Web browsing goes to privacy-infringing third-party services? That is, when a user browses on a site, what share of the HTTP request volume is not used to load the site's content from the site's servers or CDNs but to track the user's Web activity and load advertisements?

To address these questions, we identify Web services at the granularity of second level domains (SLD), i.e., multiple servers within the same SLD are considered as one service. For subdivided top level domains, e.g., uk. or au., we use the "effective SLD", as listed in the ICANN section of the public suffix list (http://publicsuffix.org/). We think that this level of granularity is best suited for this work as SLDs are most often within a single administration domain.

III. Related work

There is a large body of work on characterizing *downstream* Web traffic [1]–[7]. However, we are interested in *upstream* Web traffic. To the best of our knowledge, no previous work characterizes the upstream Web traffic of a large network as a whole.

A number of papers focus on upstream Web traffic to advertisement and analytics services. Krishnamurthy gives an overview of the related problems in an editorial [11]. Further, Krishnamurthy et al. investigate the type of information leaked in Referer headers [12] and investigate the nature of leaked information and trade-offs of possible prevention measures [13]. Roesner et al. [14] characterize the behavior of Web trackers based on data gathered with an instrumented Web browser visiting the Alexa top 500 Web sites and several embedded links. These studies mainly focus on the top advertisement and analytics services, with a strong focus on the mechanisms causing these information flows. In contrast, our approach estimates information flow in HTTP streams, independent of the technical means, and thus allows us to analyze information flow to all advertisement and analytics services. Gill et al. develop a model based on real HTTP traffic for analyzing a user's value to advertisement companies [15], but they do not analyze information flow or upstream traffic. Xia et al. [16] show that an adversary, who is recording mobile network data, can attribute up to 50 % of the traffic to users. In contrast to their work, we analyze the amount of data that different Web service classes receive.

Borders and Prakash present an approach to determine the amount of uploaded information in HTTP traffic [17]. In contrast to their approach, we strive for good scalability rather than highest accuracy, and can thus simultaneously estimate the information outflow from the traffic of many thousands of clients collected in a network or on a Web proxy server.

TABLE I: Overview of trace data. The #cIP columns show the number of IP addresses triggering Web requests, and for HTTP in parenthesis the number of IP addresses using interactive Web browsers. The duration of both traces is 24 h.

		up	down	#req.	#dom.	#cIP
HTTP	TRACE'12	61 GB	4.3 TB	63 M	103 k	23 k (18 k)
	TRACE'13	75 GB	3.4 TB	60 M	103 k	15 k (11 k)

		up	down	#con.	#dom.	#cIP
HTTPS	TRACE'12	141 GB	335 GB	5.6 M	9.3 k	20 k
	TRACE'13	213 GB	494 GB	7.4 M	9.5 k	13 k

IV. Dataset base characteristics

We base our analysis on two 24 h network packet traces recorded at the upstream router of a university campus network. TRACE'13 has been recorded during semester holidays in August 2013, and TRACE'12 in October 2012 during the lecturing period. We use the BRO IDS [18] with a custom policy to extract statistics and HTTP streams from the packet traces, and we rely on Bro for decoding Content-encoded HTTP messages. To reduce memory requirements, HTTP requests larger than 500 MB are cropped during decoding, this affects two requests of size 630 MB and 1.6 GB in TRACE'13 and one request of size 1.1 GB in TRACE'12. We restrict our analysis to connections initiated by clients from the university campus to TCP ports 80 and 443. Assuming that most Web services are using these default ports [1], we capture the vast majority of the upstream traffic to the Web.

As first step towards understanding the dataset, we investigate how many different IP addresses initiated HTTP and HTTPS connections and report our findings in Table I. The number of IP addresses is higher in TRACE'12 than in TRACE'13 because of the higher number of students during the lecturing period. In order to answer how many IP addresses are used for interactive Web browsing, we examine the User-agent header in HTTP requests. We find that between 4 k to 5 k IP addresses in each trace are exclusively using User-agents not related to interactive browsing, e.g., software updates, Web downloaders, and specialized machine-to-machine communication. We assume that the corresponding hosts are either servers, or simply have been idle. This leaves us with 11 k (18 k) IPs addresses with interactive HTTP clients in TRACE'13 (TRACE'12). Moreover, as part of our approach relies on analyzing the relationship between Web sites, we check if client IP addresses with interactive User-agents suppress Referer headers. We find that, indeed, 1.4 k (1.6 k) in TRACE'13 (TRACE'12) of the client IPs with interactive User-agents never send a Referer, however these clients only account for 486 k (1 M) requests.

In HTTPS traffic, we cannot inspect the encrypted HTTP messages. However, the SSL/TLS handshake still reveals the domain that is contacted by a browser if the Server Name Indication extension (RFC 6066) is in use. This is the case

for 75.5 % (78.3 %) of all TCP port 443 connections, covering 85.7 % (89.5 %) of the uploaded traffic volume in TRACE'13 (TRACE'12). We limit ourselves to these connections. Additionally, we check if there are dedicated SSL VPN services among the services accounting for most upstream volume. This is not the case for the top 30 SSL/TLS SLDs in each dataset. We find that HTTPS traffic is targeted to an order of magnitude fewer domains than HTTP traffic (see Table I), yet the upstream volume is about 2 to 3 times larger.

V. DIGGING INTO HTTP AND HTTPS REQUESTS

In this section, we take a closer look at upstream HTTP and HTTPS traffic. We report results only for TRACE'13, and use TRACE'12 to validate results and to peek into the past.

A. Estimating information contained in HTTP requests

The number of uploaded bytes is not necessarily a good indicator for the information disseminated by a Web client. *(i)* HTTP is known to be highly redundant [5], [10], and *(ii)* some of the uploaded data, e.g., in URLs and Cookies, have initially been sent by the server in the first place and are only reflected by the client. Therefore, to enable characterizing data flows to services more precisely, we need to find a way to identify retransmitted and reflected byte patterns in HTTP requests. Note that we focus on HTTP traffic for this part of our analysis, as we have no access to the encrypted payload of HTTPS connections. Still we point out that an organization with a TLS-intercepting proxy server can apply our methodology to HTTPS traffic too.

The sheer amount of upload traffic in our dataset unfortunately prevents us from accurately estimating the information content of HTTP requests with an information theoretical approach [17] in reasonable time. Instead, we have to rely on a better scaling approach. We eliminate redundant parts in HTTP requests by utilizing bytestring caches which we call *token caches*. A token cache is a lookup table for bytestrings observed in HTTP traffic. A token cache can be used in two ways, *(i)* to identify bytestrings that appeared in previous requests and *(ii)* to find bytestrings that have initially been provided by a server and are only reflected by a client, e.g., Cookie values. We use different token caches for every SLD and define *information flow* as the sum of the lengths of non-matching tokens. To clearly distinguish between raw upstream volume (all tokens) and information flow volume (only non-matching tokens), we use the term *information bytes* to refer to the latter. Note that our approach is a heuristic and not an exact quantification of information flow. Particularly a malicious party may hide information by encoding data in the presence or ordering of tokens, or in the timing of requests. However, unlike more exact approaches [17], the low computational complexity and limited space requirements of our method scale for the analysis of large data sets.

Our heuristic uses different separators to split HTTP requests into tokens. We split *request URLs* on characters typically used to separate directories and URL key-value parameters. For example, if the URL is *http://www.example.com/path/index.html*, then the tokens are `http:`, `www.example.com`, `path`, and `index.html`. A subsequent request to *http://www.example.com/path/picture.jpg* would find only one new token: `picture.jpg`. For multipart and urlencoded *request bodies*, we treat each form element, e.g., the e-mail address in a form or an uploaded file, as one token. The field name and value in every *request header line* are treated as separate tokens. *Cookies* are further split into key-value pairs. For *Referer headers*, we apply the same approach as for URLs. To recognize tokens that have initially been sent by the server and are reflected by the client, we also populate the token caches by extracting parts of server responses, such as Cookies, as well as URLs from `src` and `href` attributes. To improve matching for dynamically composed URLs, we additionally extract string constants from HTML and JavaScript objects.

B. Redundancy in HTTP requests – raw bytes vs. information

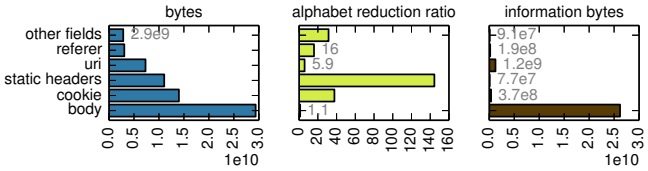
1) Overall redundancy: When applying our approach to TRACE'13 (TRACE'12), the total request volume is reduced by a factor of 2.4 (3.6), corresponding to a redundancy of 58 % (72 %). The low overall redundancy is a consequence of a limited number of large body uploads which hardly exhibit redundancy. Excluding requests bodies, the request volume is reduced by a factor of 19.3 (17.0), corresponding to a redundancy of 95 % (94 %). To get an intuition on the achieved reduction, we compare to Borders and Prakash [17]. Their information theoretic approach achieves a volume reduction to 1.48 % of the original size using a data set lacking large uploads. Our approach reduces the volume to less than 6 % when ignoring bodies, or about 4 times less. But our heuristic requires 120 times less analysis state. This allows to scale the analysis to terabytes of HTTP traffic recorded on a network gateway, while Borders and Prakash analyze the information flows of single clients, focusing on accurate information flow measurement in the presence of covert channels.

2) Information per client: TRACE'13 contains HTTP traffic of 15 k clients that communicated with 103 k Web services. In the median, a client transmits 520 kB raw bytes accounting for 32 kB information bytes per day, corresponding to a median of 32 information bytes per request.

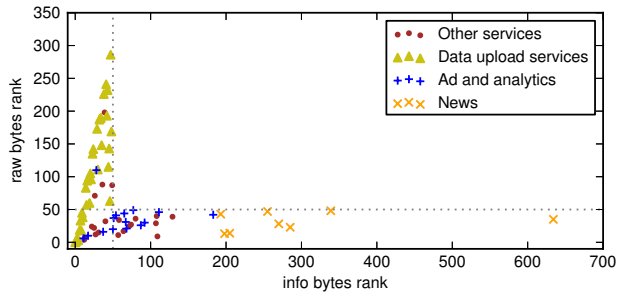
3) Redundancy in HTTP request parts: Next, we scrutinize where redundancy in HTTP requests comes from. Figure 1a shows the achieved reduction ratios for TRACE'13. We differentiate between the Request-URI, the Cookie and the Referer headers, and the HTTP body. HTTP header fields that usually stay constant for many requests (User-agent, Accept, Accept-language, Accept-encoding, Accept-charset, Connection) are summarized to the category *static headers*. Header fields not falling into any of the previously mentioned categories are accounted in the group *other fields*.

Large HTTP request bodies hardly show redundancy. This is most likely because they mainly consist of file uploads and users will rarely upload a file multiple times to the same Web service. Smaller bodies show some redundancy, e.g., requests using the Online Certificate Status Protocol (OCSP) are often identical. Such redundant OCSP requests, typically targeted at *digicert.com*, *verisign.com*, or *thawte.com*, contain a body of length 115 bytes.

In contrast, HTTP headers show higher reduction ratios in the range of 15 to 150, depending on the type of header. URIs are reduced fairly well too, yet by far not as good as



(a) Size reduction of HTTP request parts by field type.



(b) Top 50 Web services ranked by HTTP request volume measured in raw bytes (y-axis) and information bytes (x-axis).

Fig. 1: Effect of removing redundancy on TRACE'13.

static headers or Cookies. Similarly, the reduction ratio of Referer headers is limited to 16. One reason is that we use one token cache per Web service, thus URLs downloaded from one service do not reduce the accounted information flow for a subsequent request to a different service. This is intentional: Bytestrings flowing from one Web service to another actually transport information.

The next version of HTTP, HTTP/2, reduces redundancy by applying header compression [9], which has similarities to our approach. Still, we expect considerable redundancy in HTTP/2 requests too, because HTTP/2 *(i)* does not consider reflected bytestrings and *(ii)* reduces redundancy over connections of multiple users only for a small set of keywords using a static table (similar to *static headers* in Figure 1a).

C. The effect of redundancy reduction on service ranking

Next, we investigate how reducing redundancy affects Web service ranking. Figure 1b shows a rank plot including the union of the top 50 Web services for each category ranking according to raw byte request volume vs. information byte request volume. 21 services have a raw byte and information byte rank in the top 50. The plot shows that there is a clear trend regarding the service types with largest difference in rank. The highest average compression and decrease in rank exhibit eight news and news-related Web services (e.g., *spiegel.de*, *nytimes.com*, *bild.de*, *20min.ch*). An explanation for the high redundancy of this category is that while these services receive many requests and therefore have a relatively high raw request volume, the pages of these content-centric sites are relatively static as compared to sites that offer personalized content, such that requests for embedded content are mostly the same for the period of time we covered with our dataset. Thus, these services receive many raw bytes because of users browsing but only few of these bytes contain

TABLE II: Upstream Web traffic in TRACE'13. *User centric services* serve highly customized contents, often even created by users (e.g., *google.com*, *facebook.com*). *Content centric services* are less customized Web sites, showing the same contents to many users, e.g., news. *Software* stands for Web services of software vendors. 6.7 % of the HTTP requests contain a body.

		HTTP bytes	HTTP info. bytes	HTTPS bytes
Upstream volume		68 GB	28 GB	213 GB
In request headers		57 %	7.0 %	-
In request bodies		43 %	93 %	-
Top 10 client-service pairs		39 %	81 %	42 %
Top 30 services	Cloud storage	28 %	64 %	61 %
	Science-related	13 %	20.3 %	1.1 %
	User centric	11 %	2.4 %	28 %
	Advert. and analytics	4.5 %	0.7 %	0 %
	Content centric	1.5 %	0 %	0 %
	Static content	0.8 %	0.2 %	3.6 %
	Software	2.1 %	1.0 %	0.9 %
	Not classified	39 %	9.6 %	4.9 %

actually new information. The category *data upload services*, in which we group services with more than 50 % of their bytes in the body of requests, i.e., services receiving many or huge data uploads, shows the largest increase in rank. This is consistent with our insights from the previous analysis, where we observed that data uploads have only little redundancy. The 15 Web services in the category *advertisement and analytics* exhibit an average compression factor on the middle ground. *meetrics.net*, an analytics service monitoring advertisement placing is an outlier achieving only a compression factor of 1.4. In summary, we conclude that popular news and news-related Web services receive compared to other services significantly fewer information than one would assume based on raw byte counts.

D. Web service classes receiving most upstream traffic

In this section, we analyze which Web services receive most information and show that only very few clients cause these information flows.

1) HTTP traffic: Table II shows upstream HTTP traffic measured in bytes and information bytes. Two classes of Web services clearly stand out: cloud storage, and science-related. "Science" covers Web services of universities, laboratories, and publishing and management services, such as *elsevier.com*, *acs.org* and *rsc.org*. But in fact, there are only very few clients causing the dominant information flows to these classes. In TRACE'13, the top 10 client-service pairs already account for about 40 % of all bytes and 80 % of all information bytes. Closer inspection of information bytes shows that 10 different client IPs are present in these top pairs. Five clients cause traffic to network drives, two clients generate machine-to-machine traffic to scientific services, two clients upload data to the same scientific Web site, and one client to YouTube. To sum up, "cloud storage" and "science" receive so much

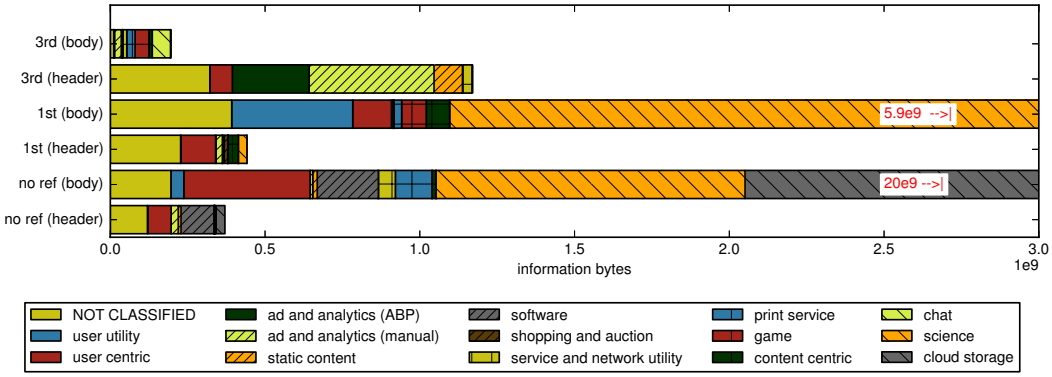


Fig. 2: Information flow in HTTP request headers and bodies per service category for TRACE'13. Two bars have been cut off, the white labels show their total sizes.

information because 9 out of the 10 heavy hitter pairs fall into these categories.

In TRACE'12, the top 10 client-service pairs account for more than 50 % of information bytes, thus the distribution is less skewed. In summary, 50 to 80 % of the information flow is caused by only 10 client-service pairs.

2) HTTPS traffic: Upstream byte counts for HTTPS traffic are shown in the right column in Table II. Cloud storage is again the category receiving most upstream volume, mainly because of the heavy-hitter domain *dropbox.com* (with 57 % of all bytes). The service *google.com* (with 22 % of all bytes) causes *user centric* to become the second most popular class. *dropbox.com* (64 %) and *google.com* (12 %) also dominated in TRACE'12.

The ten heavy hitter domains in TRACE'13 (*dropbox.com*, *google.com*, *amazonaws.com*, *vimeo.com*, *facebook.com*, *live.com*, *akamaihd.net* and three cloud storage services that are only visited by few users) receive more than 90 % of the outgoing HTTPS bytes. Similar to HTTP, only very few users cause these uploads: The listed services receive 50 % of the upstream volume from less than 1 % of the users. Exceptions are *facebook.com* and *akamaihd.net* with 3 % and 11 % of the users causing 50 % of the uploads, respectively. The top 10 client-service pairs cause 42 % (36 %) of the upstream HTTPS traffic in TRACE'13 (TRACE'12).

3) Summary: We conclude that very few clients and services have a large impact on upstream Web traffic, similar to downstream Web traffic [8]. We note that this trend is still present if science-related services, a prominent service class in the analyzed university network, are not considered.

E. Investigating information to first- and third-party Web services

Nowadays Web pages are a conglomeration of dozens to hundreds of embedded objects loaded from different dedicated services [19], such as content distribution networks, social network plug-ins, video portals, as well as advertisement and analytics services. As a result, a user visiting a Web page causes not only HTTP requests to the visited Web service—the first party—but, as a side effect, also many requests to these third parties. Hence, we analyze how much information is leaking to first- and third-party services. We can distinguish between first- and third-party requests by comparing the domain in the request URL to the domain in the Referer field in the request header. If they are the same, it is a first-, otherwise a third-party request. As third case, the Referer may not be set at all. For example, if a resource is accessed without following a link, if the browser is configured to suppress Referer headers, if the original page is loaded over HTTPS but the current request is not, or if the resource is accessed by special software instead of a Web browser. In Section V-B we found that request bodies dominate headers with respect to transmitted information. Therefore, in order to identify services only prevalent in headers, we analyze headers and bodies separately.

For this analysis, we need access to the HTTP request header to tell apart first- from third-party requests, therefore we focus on HTTP traffic. Figure 2 shows a break-down of information flow in TRACE'13. We manually classify the top 30 services in each category and rely on blacklists from Adblock Plus[2]. The resulting classification, which consists of 13 service classes, covers 95 % of the outgoing HTTP information flow.

Special software was used for the cloud storage services and some of the science-related data transfers, explaining why no Referer header is set for these transfers. The class user centric services, in which we summarize services highly customized for every user, e.g., *youtube.com*, *google.com*, *facebook.com*, *twitter.com*, ranks highest in the category *header – first party* because of a large share of users interacting with these services. The classified content centric Web pages (e.g., *theguardian.com*, *bild.de*, *20min.ch*) only account for 8 % of information flow in this category. User utility services, i.e., Web tools for data and document processing, particularly online PDF editing, mainly appear as first parties; probably because users visit these Web sites and upload documents using a Web form.

Web services which are related to software vendors, e.g., *sophosxl.net*, *apple.com*, *avira-update.com*, *windowsupdate.com*, mainly appear in requests without Referer because

[2]https://adblockplus.org/

these services are often accessed by software of the corresponding vendors. The service *sophosxl.net* registered by Sophos is of particular interest because it receives 23 % of all information flow in headers. *sophosxl.net* receives 20 times more information than the second most popular anti-virus update service *avira-update.com*. Closer inspection reveals that Sophos Endpoint Security is installed on many clients in the analyzed network and reports URLs that are visited by clients to this domain to check for malware. It has been reported that also URLs visited over HTTPS are transmitted in plain HTTP requests to this service only protected by a ROT13 obfuscation [20].

Static content services (e.g., *gstatic.com*, *akamaihd.net*, *googleapis.com*) receive mainly information in headers as third parties. Few clients using a video chat Web site that load-balances chats over different servers cause chat services to become the top class in the category *third party - body*. Browser games seem to transmit information in request bodies. This class appears not only as first but also as third party because a popular browser game is played from within the Web site of a game publisher.

Advertisement and analytics services (e.g., *doubleclick.net*, *google-analytics.com*, *meetrics.net*, *criteo.com*, *revsci.net*) mainly appear in the category *third party - header* and clearly dominate this category. The distribution of services is less skewed in this category than in others, which is why our manual classification of the top 30 services only covers half of the information flow. To get an intuition how much more information is related to advertisement and user tracking, we additionally check if third-party services are listed by the popular anti-tracking tool Adblock Plus (ABP) [3]. We find additional 1.8 k services that are blacklisted by ABP (labeled as *ad and analytics (ABP)* in Figure 2) and 4.5 k services, for which their SLD appears as substring in a blacklisted domain.

F. Understanding the dimension of Web tracking

In order to quantify the dimension of Web tracking taking place, we estimate what share of the HTTP request volume is not used to load the visited site's content but goes to advertisement and analytics services. We focus on HTTP traffic because we need to distinguish between first- and third-party requests. More specifically, we focus on HTTP request headers since most information flow during Web browsing happens in HTTP request headers and not in HTTP entity bodies. Bodies are mainly used to upload data and are heavily dominated by uploads of very few clients (see Section V-D). Further we exclude the information flows to the class "science" from this analysis, because these Web sites are specific to university networks. This allows us to make a more general statement. We argue that information bytes is a better approximation to quantify the amount of tracking than counting the number of raw bytes because redundancy in requests differs significantly between service classes (see Section V-C). Still, we point out that counting information bytes is only an approximation for the dimension of Web tracking taking place because not all information transmitted to advertisement and analytics services might actually be used for tracking.

[3]We use the Adblock Plus filters EasyList (advertisement) and EasyPrivacy (analytics) and a specific EasyList for our region, all available on https://easylist.adblockplus.org/en/

We start with a conservative estimation: Figure 2 shows that advertisement and analytics services receive about 650 MB of information as third parties (56 % of all third-party information bytes) and additional 22 MB as first parties. All first-party information flows (440 MB) without the ones to advertisement and analytics (22 MB) and science services (28 MB) account for about 390 MB. This means that advertisement and analytics services receive about 1.7 times more information than all first-party services together.

For the above estimation, we considered manually labeled advertisement and analytics services and services, for which the complete SLD is blacklisted. If we also consider SLDs that appear as substring in a blacklisted domain, we find that about 790 MB of information bytes go to corresponding third-party services (67 % of all third-party information bytes). But also multiple user centric sites, e.g., *facebook.com*, *twitter.com*, *google.com* provide some kind of "like"-buttons. This allows them to track users similarly to advertisement and analytics services. The user centric service has in this case the role of a third party. Indeed, the second bar in Figure 2 shows 73 MB of information flowing to user centric sites in the role of third parties. If we recalculate with these numbers, we arrive at a factor of 2.3 instead of 1.7. We conclude that advertisement services and services related to user tracking receive about 2 times more information than all first-party services together.

VI. Conclusion

The aim of our work is to understand the upstream Web traffic of a real network. For this purpose, we investigate the entire upstream Web traffic of a large campus network. Our study is guided by two key questions. First, we ask which classes of Web services receive most non-redundant information. Second, we quantify the share of information going to privacy-infringing third-party services (such as trackers) during Web browsing.

HTTP traffic between individual users and Web services usually contains a high amount of redundancy. Therefore, we propose a heuristic to remove most of this redundancy in outbound HTTP traffic in order to estimate the amount of information transmitted to Web services more precisely. Our heuristic significantly reduces the upstream volume of the studied network traces by a factor of 2.4 to 3.6 for entire HTTP requests and by a factor of 17 to 19 for HTTP request headers.

To answer the first key question, we find two main modes of operation: *(i)* large information chunks generated by typically only very few users, e.g., for network drives or paper submission sites, and *(ii)* Web services receiving little pieces of information from a huge number of requests and a large user base. The services *dropbox.com* and *google.com* clearly receive most upstream information according to mode *(i)*. Additionally, science-related Web services rank high if only considering uploads over HTTP. Altogether, we classify up to 95 % of all information flow to first- and third-party HTTP services.

To answer the second key question, we find that advertisement and analytics services receive about 60 % of all information flowing to third parties and about two times more (non-redundant) information than all first-party Web services. In our evaluation, we exclude upstream traffic events dominated by

few clients and science-specific Web sites. Hence, this finding could also apply to other networks.

ACKNOWLEDGEMENT

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

REFERENCES

[1] G. Maier, A. Feldmann, V. Paxson, and M. Allman, "On dominant characteristics of residential broadband internet traffic," in *Proc. IMC '09*, 2009. [Online]. Available: http://doi.acm.org/10.1145/1644893.1644904

[2] M. E. Crovella and A. Bestavros, "Self-similarity in world wide web traffic: evidence and possible causes," *IEEE/ACM Trans. Netw.*, vol. 5, no. 6, Dec. 1997. [Online]. Available: http://dx.doi.org/10.1109/90.650143

[3] M. F. Arlitt and C. L. Williamson, "Web server workload characterization: the search for invariants," in *Proc. SIGMETRICS '96*, 1996. [Online]. Available: http://doi.acm.org/10.1145/233013.233034

[4] P. Barford, A. Bestavros, A. Bradley, and M. Crovella, "Changes in web client access patterns: Characteristics and caching implications," *World Wide Web*, vol. 2, no. 1-2, 1999. [Online]. Available: http://dx.doi.org/10.1023/A%3A1019236319752

[5] S. Ihm and V. S. Pai, "Towards understanding modern web traffic," in *Proc. IMC '11*, 2011. [Online]. Available: http://doi.acm.org/10.1145/2068816.2068845

[6] 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*, 1997.

[7] M. Butkiewicz, H. V. Madhyastha, and V. Sekar, "Understanding website complexity: Measurements, metrics, and implications," in *Proc. IMC '11*, 2011. [Online]. Available: http://doi.acm.org/10.1145/2068816.2068846

[8] F. Schneider, B. Ager, G. Maier, A. Feldmann, and S. Uhlig, "Pitfalls in http traffic measurements and analysis," in *Proc. PAM '12*, 2012. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-28537-0_24

[9] R. Peon and H. Ruellan, "HPACK - Header Compression for HTTP/2." [Online]. Available: https://tools.ietf.org/html/draft-ietf-httpbis-header-compression-12

[10] A. Anand, C. Muthukrishnan, A. Akella, and R. Ramjee, "Redundancy in network traffic: findings and implications," in *Proc. SIGMETRICS '09*, 2009. [Online]. Available: http://doi.acm.org/10.1145/1555349.1555355

[11] B. Krishnamurthy, "I know what you will do next summer," *SIGCOMM Comput. Commun. Rev.*, vol. 40, no. 5, Oct. 2010. [Online]. Available: http://doi.acm.org/10.1145/1880153.1880164

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

[13] 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*, 2007. [Online]. Available: http://doi.acm.org/10.1145/1280680.1280688

[14] F. Roesner, T. Kohno, and D. Wetherall, "Detecting and defending against third-party tracking on the web," in *Proc. NSDI '12*, 2012. [Online]. Available: http://dl.acm.org/citation.cfm?id=2228298.2228315

[15] 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*, 2013. [Online]. Available: http://doi.acm.org/10.1145/2504730.2504768

[16] N. Xia, H. H. Song, Y. Liao, M. Iliofotou, A. Nucci, Z.-L. Zhang, and A. Kuzmanovic, "Mosaic: Quantifying privacy leakage in mobile networks," in *Proc. SIGCOMM '13*, 2013. [Online]. Available: http://doi.acm.org/10.1145/2486001.2486008

[17] K. Borders and A. Prakash, "Quantifying information leaks in outbound web traffic," in *Proc. SP '09*, 2009. [Online]. Available: http://dx.doi.org/10.1109/SP.2009.9

[18] V. Paxson, "Bro: a system for detecting network intruders in real-time," *Computer Networks*, vol. 31, no. 23-24, 1999. [Online]. Available: http://dx.doi.org/10.1016/S1389-1286(99)00112-7

[19] 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)*, 2012. [Online]. Available: http://dx.doi.org/10.1109/NGI.2012.6252145

[20] Portcullis Labs, "Could sophos antivirus web protection cause a privacy concern for your organisation?" [Accessed: 2015-02-25]. [Online]. Available: https://labs.portcullis.co.uk/blog/could-sophos-antivirus-web-protection-cause-a-privacy-concern-for-your-organisation/