Towards web service classification using addresses and DNS

Towards Web Service Classification using Addresses and DNS

Martino Trevisan, Idilio Drago, Marco Mellia, Maurizio M. Munafò
Politecnico di Torino, Italy
(firstname.lastname@polito.it)

***Abstract*—The identification of the services that generate traffic is crucial for ISPs and companies to plan and monitor the network. The widespread deployment of encryption and the convergence of the web services towards HTTP/HTTPS challenge traditional classification techniques. Algorithms to classify traffic are left with little information, such as server IP addresses, flow characteristics and queries performed at the DNS. Moreover, due to the usage of Content Delivery Networks and cloud infrastructure, it is unclear whether such coarse metadata is sufficient to differentiate the traffic. This paper studies to what extent basic information visible at flow-level measurements is useful for traffic classification on the web. By analyzing a large dataset of flow measurements, we quantify how often the same server IP address is used by different services, and how services use hostnames. Our results show that a very simple classifier that relies only on server IP addresses and on lists of hostnames can distinguish up to 55% of the traffic volume. Yet, collisions of names and addresses are common among popular services, calling for more ingenuity. This paper is a preliminary step in the evaluation of classification algorithms that are suitable for the modern Internet, where only minimal metadata collection will be possible in the network.**

I. Introduction

Monitoring how web services are used and how they consume network resources is key to Internet Service Providers (ISP) when operating and planing the network. Similarly, companies have a vital need of monitoring their enterprise networks – e.g., to ensure usage of accredited services, or to control the access to unauthorized ones. Traffic classification has always taken a key role, and a variety of methods has been developed throughout the years. Initially focusing on protocol classification, e.g., HTTP vs FTP vs P2P, classification goals must now target the identification of "web services", e.g., YouTube vs Facebook vs Whatsapp. Indeed, HTTP is becoming the de-facto application layer protocol over which people access the large majority of Internet applications. Deep Packet Inspection (DPI), behavioral techniques [1], [2], have been used for traffic classification. These methods have been recognized so far as effective for several monitoring needs [3].

The convergence of web toward proprietary and encrypted protocols, however, challenges classification algorithms again. Indeed, we already observe a clear trend towards moving Internet services to protocols such HTTPS [4], with HTTP 2.0 behind the corner and TLS encryption by default. While this trend is well-justified by the urgency in improving end-users' privacy, it renders many traffic classification algorithms useless, since packet payload cannot be accessed anymore.

In addition, a handful of big players [5] is taking a prominent role in the Internet, where content is more and more being served from shared infrastructure, such as in Content Delivery Networks (CDNs) and cloud computing platforms. This further challenges behavioral classifiers [6], which rely on host profiling to determine the applications running on servers.

This paper revisits the question of whether basic traffic features can be used to differentiate traffic of major web services. The ambitious goal is to understand how feasible would be the classification of web services traffic based only on server IP addresses and queries to the DNS, i.e., the few features that are likely going to remain visible. By relying on a large dataset containing flow-level measurements of user activity annotated with DNS queries, we first investigate to what extent server IP addresses provide enough evidences of the services used by people. We then evaluate the amount of traffic that can be distinguished by combining server hostname and addresses to create rules.[1] Finally, we discuss how stable such rules are in time.

Previous works have studied the importance of different features for traffic classification. In particular, a comprehensive survey on classification methods for encrypted traffic is presented in [7]. The authors of [8], [9] found that IP addresses are among the most informative features. We perform similar analysis to quantify how traffic of modern services can be classified using only addresses and hostnames. Authors of [10] are the first to claim the use of DNS to classify traffic. In contrast to the method proposed by authors of [10], we neglect well-known protocols (e.g., FTP or P2P). Instead, we focus on typical services that make the majority of encrypted web traffic nowadays, and characterize when hostnames are needed, and when only addresses would be sufficient for classification. More recently, authors of [11], [12] used Server Name Indication (SNI) strings found in TLS handshakes and DNS queries for classification. While authors concluded that hostnames alone are insufficient, they targeted protocol classification (e.g., SIP, HTTP, etc.), thus missing fine-grained identification of single web services. Other authors [13], [14] argue the usefulness of DNS for classifiers, but mostly focusing on how to label flows, missing a study of classification accuracy.

This research has been funded by Cisco Inc. and by the Vienna Science and Technology Fund (WWTF) through project ICT15-129, "BigDAMA"

[1] In the remaining of the paper, we omit the word "server" unless necessary.

Our work is a preliminary evaluation of web service classifiers in the modern Internet. Our analysis provides the following main findings:

• Up to 65% of the IP addresses are associate to a single hostname. Those servers however are responsible for less than 15% of web traffic volume.

• Despite the simplicity, classification based solely on (group of equivalent) IP addresses can discern up to 55% of the web traffic. This can be achieved by uncovering and aggregating the various hostnames related to a given service, and then enumerating corresponding IP addresses.

• Lifetime of classification rules varies strongly, with some services requiring weekly updates and others showing stable names and addresses even after a year.

• Even when tagging flows with hostnames on-line using all DNS queries of each client (e.g., as in [13]), there can be complex scenarios when facing big cloud computing platforms (e.g., Amazon AWS)

These results are a first step towards classification algorithms that are able to work with minimal metadata. While these data will certainly not solve some identification problems (e.g., for network forensics and intrusion detection), we believe they represent a set of non-intrusive features to tackle common monitoring tasks, such as traffic accounting and engineering.

II. Datasets and Methodology

The aim of this work is to investigate whether IP addresses and DNS traffic provide enough information to design web service classifiers, targeting in particular those prominent services which adopt encryption, such as HTTPS, QUIC or SPDY. We take a data driven approach and look into real traces to run a feasibility check in this paper, before going through a complete system design.

A. Datasets

We use two data sources in our analysis. First, we rely on *Tstat* [15] to perform passive measurements and collect data related to users' activity. Among flow level statistics exposed by *Tstat* for *IPv4* TCP flows, we consider (i) server IP addresses contacted by clients; (ii) timestamps of the first packet in each flow; (iii) SNIs sent by clients in TLS handshakes; and (iv) the hostname the client resolved via DNS queries prior to open the flow.[2] This mechanism, called DN-Hunter, is explained in details in [13]. Inconsistencies are observed between SNI and DNS in the 6% of flows.

Although *Tstat* exports many other flow-level metrics that are useful for traffic classifiers, we restrict the analysis to basic features, since those features are also exported by popular flow meters such as *Netflow* [16].

Second, in parallel to *Tstat*, we deploy *Passive DNS*[3] in one of the monitored links to get a deeper insight into the association between hostnames and server IP addresses. *Passive DNS* logs all DNS activity in the network independently from the resolver the client employs, including queries and responses with the returned addresses and the time-to-live found.

[2]Our vantage points observe all traffic generated by clients, including DNS traffic directed to local resolvers.

[3]https://github.com/gamelinux/passivedns

TABLE I: Overview of our datasets.

Name	Flows	Server IPs	Period	Sections
PoP	13.25G	49.25M	1 year	VI
Campus-Flows	1.12G	2.55M	2 months	IV,V,VI,VII
Campus-DNS	–	1.13M	2 weeks	III,IV,V,VI

Table I summarizes our datasets. We have installed *Tstat* in two distinct networks: (i) a University campus in Europe where $\approx 15,000$ users are connected; and (ii) a Point of Presence (PoP) of a European ISP, where $\approx 10,000$ ADSL customers are aggregated. The campus dataset includes traffic generated by wired and WiFi networks during 2 months in 2015. *Passive DNS* was deployed in the campus for 2 weeks in Nov 2015. The residential dataset includes traffic of users' devices connected via Ethernet and/or WiFi at home during a full year (2015). In total, our datasets include statistics about more than 14 billion flows, and around 790 million records in DNS requests/responses.

B. Methodology

We study the association between IP addresses and hostnames to understand the role of addresses in modern traffic classification. We first assume hostnames provide sufficient means to distinguish services – i.e., different services use different hostnames. We will discuss later to what extent this assumption holds in practice. Hostnames coming either from SNIs or from DNS queries are the ground-truth in this scenario. We characterize how the relation between names and addresses evolves over time. In particular, we look for those IP addresses that serve only a single hostname, i.e., only one hostname is associated to a given IP address. We call this *singleton IP addresses*, or singleton in brief. We then quantify the percentage of traffic exchanged with singletons, to obtain an indication of the classification coverage that could be achieved using only the IP addresses as features.

Motivated by the low volume of traffic that could be discerned by such an approach, we study how to improve classification by enumerating the different hostnames (and addresses) used by services. We call the list of names of a service its *bag of domains*. We interactively build the bag of domains for a list of services by relying on SNIs and hostnames exported by *Tstat*. A graphical framework allows us to inspect names linked to IP addresses. We illustrate this procedure with examples in the next section. We focus on popular services running over HTTPS – e.g., Facebook, Google Video, Dropbox, Apple iCloud, Twitter etc. – since those services cause the greatest part of the encrypted traffic in the monitored networks.

III. Enumerating Names and Addresses of Services

We visually explore how hostnames and addresses are associated. We represent the associations as a graph, in which nodes are IP addresses and hostnames, and edges exist if a hostname has been resolved to an address. We initially search

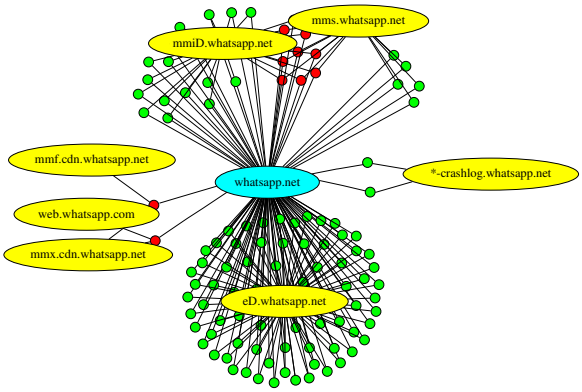


Fig. 1: IP addresses and hostnames of Whatsapp. Most IP addresses are exclusively used by the service.

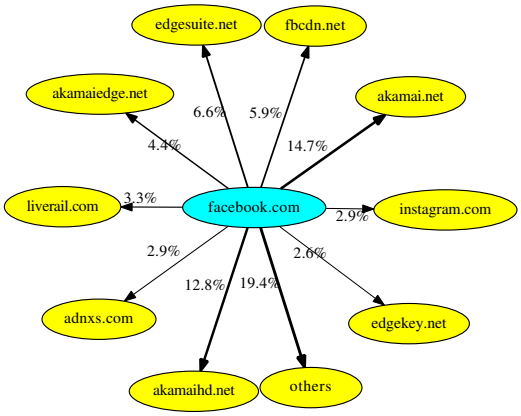


Fig. 2: Hostnames sharing IP addresses with Facebook.

for hostnames containing terms of interest. For example, by searching for `whatsapp` in our data, we discover that Whatsapp services are offered from at least two second-level domains – i.e., `whatsapp.com` and `whatsapp.net`. We call those the *core domains*, and from them we explore linked IP addresses, and correlated hostnames.[4]

Fig. 1 and Fig. 2 provide examples. Fig. 1 depicts how second-level domains are associated with `whatsapp.net`. For simplicity, the figure is built using a 5-minute sample of *Campus-DNS* trace. The core domain is shown as a central node; IP addresses are nodes colored either green (singletons, i.e., edge links them to only one `whatsapp.net` sub-domain), or red (not singletons, with multiple edges to multiple domains); and yellow nodes represent `whatsapp.net` sub-domains sharing IP addresses with each others.

We notice that Whatsapp IP addresses are not shared with other services. Therefore, once all addresses are enumerated, Whatsapp traffic can be identified without further information.

Fig. 2 shows more complicated scenarios emerging from `facebook.com`. To improve visualization, nodes representing IP addresses are replaced by edges labeled with the percentage of addresses connected to pairs of hostnames – e.g., 3.3% of the addresses seen as `facebook.com` are also seen

[4]Terms of interested could be obtained by active experimentation with target services in a testbed such as in [17].

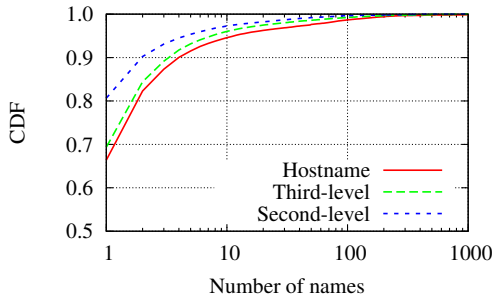


Fig. 3: Distribution of names per IP address. *Campus-DNS*.

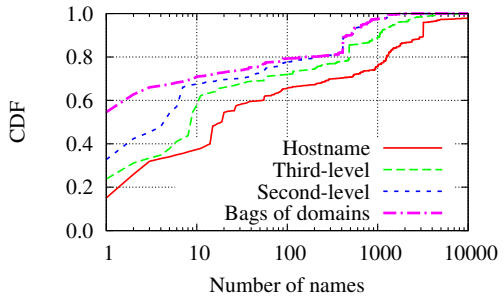


Fig. 4: Distribution of the traffic related to IP addresses with different numbers of hostnames. *Campus-Flows*.

as `liverail.com`. Besides sharing addresses with Facebook's affiliated services (e.g., Instagram), Facebook's usage of Akamai CDN results in thousands of hostnames unrelated to Facebook appearing in the graph as time progresses.

In summary, the association between IP addresses and hostnames brings information, but the presence of CDNs create conflicts and ambiguity. Next, we quantify how much traffic could be classified despite such ambiguities.

IV. Classification Using IP Addresses

We first provide an overview on all IP addresses and hostnames in our 2-week long dataset of DNS traffic (i.e., *Campus-DNS*). We perform this analysis focusing on DNS *A* records. For each IP address returned in a DNS response, we collect all hostnames requested by clients.

Using *Campus-DNS* dataset, we count how many hostnames are linked to each IP address. Three levels of granularity are considered: (i) complete hostnames, e.g., `img.www.example.com`; (ii) third-level domains, e.g., `www.example.com`; (iii) second-level domains, e.g., `example.com`. Fig. 3 reports the empirical Cumulative Distribution Function (CDF) of the number of names associated with each IP address.

On the leftmost part, we notice that more than the 65% of the IP addresses are singletons. This percentage grows to 70% when considering third-level domains, and to 80% when considering second-level domains. Those results confirm

TABLE II: Popular services and classification precision.

Core Domain	*facebook.com*	*google.com*	*googlevideo.com*	*whatsapp.net*	*twitter.com*	*dropbox.com*
Number of Addresses	3,196	7,286	13,133	851	279	2,227
Singletons for the service (%)	29.8	58.7	79.9	99.8	83.9	59.9
Traffic to singletons (%)	85.8	38.5	1.2	100.0	78.7	91.3
Precision (%)	59.1	33.8	77.2	100.0	96.1	99.1

previous observations (e.g., see [13], [18]) and, at first, suggest that a great part of the traffic could be easily classified by simply using server IP addresses.

A completely different picture however emerges when taking traffic volume into account. Although most IP addresses are singletons, such addresses are responsible for a small fraction of the traffic. We quantify this effect in Fig. 4 using the *Campus-Flows* dataset. For each IP address, we count the amount of bytes it handles, and compute then the handled fraction. Fig. 4 shows the resulting CDF. Remind that we include only HTTP and HTTPS traffic here. Less than 15% of the traffic is owing to singletons, even if those cases are 65% of the addresses. The picture does not change considerably when third- or second-level domains are used: Percentages are 25% and 33%, respectively. In a nutshell, a classifier that takes only IP addresses as input would identify up to 33% of the traffic without mistakes. Part of the remaining traffic would necessarily be misclassified, since many hostnames (and thus services) possibly run over the same address. We conclude that server IP addresses alone provide a very poor classification coverage for the web traffic.

V. Classification Using Bags of Domains

We repeat the analysis after creating *bags of domains*. A bag represents the set of domains a service uses to handle its content. We consider 25 coarsely defined groups of services, including e.g., *Google*, *Facebook* and *Dropbox*. For example, Facebook bag of domains includes `facebook.com` as well as Facebook's domains pointing to CDNs, such as `fbcdn.net` and `fbstatic-a.akamaihd.net`. So far, we manually group domain names that belong to a given service, since we observe that bags of domains are rather stable in our datasets.

Given a bag of domains, we extract IP addresses corresponding to any name in the bag. We next check if those addresses have been resolved also for hostnames not in the bag. Those IP addresses that create ambiguity are discarded. Those that correspond to hostnames in the bag only are *singletons for the service* and thus provide a good classification, i.e., traffic is uniquely linked to the targeted service.

Fig. 4 reports the CDF of traffic according to singletons for the services. The bags of domains substantially increases the fraction of traffic that can be discerned. Three regions are visible in the figure. First, close to 55% of the traffic is related to IP addresses that are connected to a single bag of domains. Second, up to 10% of the traffic is caused by IP addresses shared by at most 10 names or bags. Part of these cases seems to occur because we have created bags only for few popular services, and thus names could be aggregated further. Third, about 20% of the traffic volume is caused by IP addresses shared by hundreds or thousands of names. Those cases are mostly servers in CDNs, and it is hard to discern services without full information about hostnames queried by clients. The intuition suggests that the bag of domains approach would be ineffective for this latter group. We will investigate these cases further in coming sections. We perform a similar calculation accounting flows instead of bytes, obtaining very similar results, not reported here for lack of space.

We conclude that a very simple classifier that relies on server IP addresses only could discern up to 55% of the web traffic. However, this is achievable only if service hostnames are aggregated, and their addresses are enumerated. Important, IP addresses in bags of domains can be learned by inspecting logs in DNS servers, or by actively querying the DNS system. Finally, the development of a methodology to automatically create bags of domains and to enumerate IP addresses is planned to future work.

VI. Use Cases

A. A Deeper Look into Popular Services

We now focus on six popular services and study in details how hostnames and addresses are used. We further estimate the precision of different classification approaches when applied to these services. Tab. II reports statistics about 6 services over two weeks of observations. We calculate statistics using the period in which *Campus-Flows* and *Campus-DNS* datasets are coincident. We focus on the most popular web services categories such as Social Networks, Search Engines and Cloud Storage. Thus, we take into account Facebook, Google, Whatsapp, Dropbox and Twitter, considering all traffic to their *bags of domains*.

Tab. II shows that the number of IP addresses hosting each service (2nd row) varies considerably,[5] as it varies the percentage of those addresses that are fully dedicated to the services (3rd row - singletons). For instance, while 99.8% of the IP addresses serving Whatsapp are singletons, more than 40% of the addresses of Google are observed in DNS queries related to non-`google.com` bag of domains. No address has been seen in more than one of the considered bags, except for Google and Googlevideo: all non-singletons of Google Video appear within Google's bag, and the 89.9% of Google's are in Google Video's, unveiling a shared infrastructure.

Next, we quantify the traffic related to singletons (4th row): Using the *Campus-Flows* trace, and using the DN-Hunter or SNI as ground truth to identify the service associated to a flow, we sum up all traffic for all hostnames in each bag of domains. We then compute the fraction of traffic that is associated with

[5]The total number of addresses serving each service is likely higher since only *contacted* addresses are counted.

singletons for the same service. This number gives us an estimation of the coverage if one relies only on the singletons to classify – i.e., the coverage when the classification provides 100% precision.

We can see that the percentage of traffic going to singletons is quite low for some services. Note for instance that only 1.2% of Google Video traffic goes to singletons, despite these being almost 80% of IP addresses of `googlevideo.com`. This happens since the traffic balance among the thousands of GoogleVideo servers is highly skewed toward a small subset of them, i.e., the most popular ones. Those addresses are also the ones for which hostnames of other bags of domains are found, and thus they are not singletons. For other services, singletons provide very high coverage: 100% of Whatsapp traffic goes to singletons (cfr. Fig. 1), whereas percentages are relatively high also for Facebook (85%), Twitter (78%) and Dropbox (91%).

Finally, we estimate the precision of a classifier that marks *all* traffic related to addresses in the bags as belonging to the given services, being those singletons or not. That is, we estimate the precision of a classifier that have maximum coverage for the selected services. Since not all addresses are singletons (see 2nd row in Tab. II), we expect to make classification mistakes.

The last row in Tab. II quantifies such mistakes. We can see that for three examples in the table – Whatsapp (100%), Twitter (96%) and Dropbox (99%) – the precision would be indeed very high. This means that only a minor amount of traffic not belonging to the services is mixed in their bags of domains. Google Video also presents a very high precision thanks to high traffic volume of the YouTube video service. Services that are mixed up with Google Video produce a low volume, even if they reach addresses in the Google Video bag. For Facebook, the classification precision is rather low, and this questions the applicability of the approach for such cases. This is because Facebook uses Akamai CDN, which hosts a multitude of alien services, which generate overall a large amount of traffic.

All in all, the classification based solely on addresses and bags of domains shows interesting potential. It enables the classification of a high share of traffic, with high coverage and precision for many popular services, while requiring minimal collection of data. Yet, a per-service assessment of precision and coverage is needed.

B. Names and Addresses over Time

We analyze how the associations between names and addresses evolve over time. In particular, we are interested in knowing how stable the rules based on IP addresses and bags of domains are for popular services. We investigate such aspects using *PoP* dataset, which covers a full year of a residential network. For each month of data, we create lists with all addresses used by popular services considering their bags of domains. We then track how the lists change throughout the year.

Fig. 5 summarizes results by showing the percentage of addresses that is still on the lists, when compared to the first month of observation. We can see that all services present

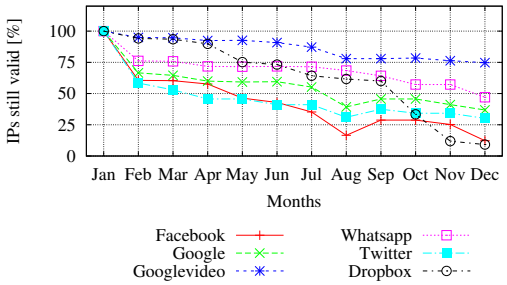


Fig. 5: Persistence of addresses of popular services.

changes after Jan 2015, which is used as reference in the figure. Similar shapes emerge if other months are taken as reference. However, it is interesting to notice differences among services. Whereas the list of addresses for Google Video, for instance, is rather stable, as little as 15% of the Dropbox addresses seen in Jan 2015 remain in the list. Manual inspection suggests that addresses are passing for migration from US data-centers to EU data-centers; clients are now diverted to different data-centers than in previous months [6].

In several cases, such as for Twitter, almost 50% of the addresses already disappeared after a single month of observation.

Overall, we conclude that while the lists of addresses are stable in short intervals, they radically change in medium to long periods (Fig. 5). Such intervals strongly depend on services and location of the monitored network. Classifiers relying on lists of addresses must deploy a methodology to constantly check and update their lists.

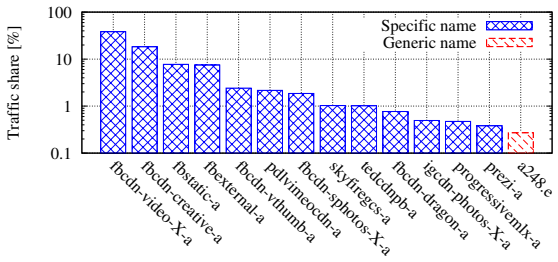
VII. Traffic in Ambiguous Names

In the previous sections we evaluated the traffic related to IP addresses using the annotated hostnames as ground-truth. Now we investigate to what extent annotated traffic is reliable to the classification problem. We evaluate the case where each flow is associated to a hostname directly at the vantage point, as done by DN-Hunter or by extracting the SNI via DPI. Thus, the question is whether hostnames are unique to bags of domains of different services. We thus quantify how often hostnames are used by different services.

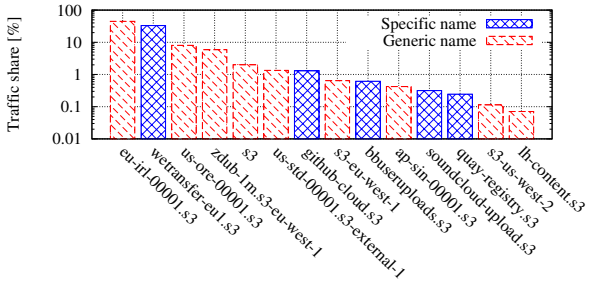
We focus on two examples, *Amazon Web Services (AWS)* and *Akamai*, and enumerate all sub-domains of `amazonaws.com` and `akamaihd.net` contacted by clients in our datasets. Then, we manually try to identify the services relying on each sub-domain. In some case, hostnames give a clear hint about services – e.g., `fbcdn-sphotos-c-a.akamaihd.net` is used by Facebook, although generic names of the infrastructure providers are often observed as well – e.g., `eu-irl-00001.s3.amazonaws.com` is used by many services outsourcing to AWS.

Fig. 6 highlights the top sub-domains of the providers according to their traffic share. Sub-domains are split into two groups: specific and generic. The first contains sub-domains

[6]See also https://www.dropbox.com/help/9063



(a) Akamai



(b) AWS

Fig. 6: Top sub-domains hosted by Amazon and Akamai.

that can be definitively associated to a service, whereas all other sub-domains are marked as generic.

When summing up all bytes related to specific sub-domains, we notice that 98% of the traffic related to Akamai can be distinguished. Therefore, classifiers can reach high coverage and precision when handling Akamai traffic, provided that information about hostnames requested by clients is available.

The scenario is different for AWS. Only 23% of the traffic related to Amazon has an informative sub-domain. We can see in Fig. 6 that only 3 among the top-14 AWS sub-domains in our datasets provide hints on the service generating the traffic. Such cases without informative names are indeed hard to be discerned and will require a much more elaborated classification methodology. We plan to test in future work classifiers that correlate names of distinct flows, including both temporal and spacial correlations among flows.

VIII. Conclusions

This paper provided a first look into traffic classification for modern web services. We visually explored how hostnames and addresses are associated, and studied the role of IP addresses in classification. Our results show that up to 55% of web traffic can be identified relying solely on addresses. This coverage is however achieved only if the several hostnames used by services are uncovered, and the respective addresses are enumerated. For some specific services, IP addresses can classify most of the traffic. Those results call for the development of novel classification methods, which will operate with minimal information collected from the network, thus respecting users' privacy.

Nevertheless, we also pointed out that the association between hostnames and addresses changes frequently. For instance, for a selection of services, more than half of the addresses were changed during one year of observations.

This paper identified several directions for future work. Firstly, we showed that a number of services shares hostnames, in particular those services hosted at cloud environments. The identification of services is not possible in such cases, even when flows are tagged with client-requested hostnames. Methods to classify this traffic are needed, and we will pursue that in the future. Secondly, we plan to design and implement algorithms to automatically retrieve the list of hostnames associated to services (i.e., the bags of domains) as well as to detect changes and to update the list over time. Finally, we plan to integrate these techniques into a complete system to detect and account traffic of popular services on real-time.

References

[1] H. Kim, K. C. Claffy, M. Fomenkov, D. Barman, M. Faloutsos, and K. Lee, "Internet Traffic Classification Demystified: Myths, Caveats, and the Best Practices," in *Proceedings of the CoNEXT*, 2008, pp. 1–12.

[2] A. Callado, C. Kamienski, G. Szabó, B. P. Gero, J. Kelner, S. Fernandes, and D. Sadok, "A Survey on Internet Traffic Identification," *Commun. Surveys Tuts.*, vol. 11, no. 3, pp. 37–52, 2009.

[3] S. Valenti, D. Rossi, A. Dainotti, A. Pescapè, A. Finamore, and M. Mellia, "Reviewing Traffic Classification," in *Data Traffic Monitoring and Analysis - From Measurement, Classification, and Anomaly Detection to Quality of Experience*, 1st ed. Heidelberg: Springer, 2013.

[4] D. Naylor, A. Finamore, I. Leontiadis, Y. Grunenberger, M. Mellia, M. Munafò, K. Papagiannaki, and P. Steenkiste, "The Cost of the "S" in HTTPS," in *Proceedings of the CoNEXT*, 2014, pp. 133–140.

[5] V. Gehlen, A. Finamore, M. Mellia, and M. M. Munafò, "Uncovering the Big Players of the Web," in *Proceedings of the TMA*, 2012, pp. 15–28.

[6] T. Karagiannis, K. Papagiannaki, and M. Faloutsos, "BLINC: Multilevel Traffic Classification in the Dark," *SIGCOMM Comput. Commun. Rev.*, vol. 35, no. 4, pp. 229–240, 2005.

[7] P. Velan, M. Čermák, P. Čeleda, and M. Drašar, "A Survey of Methods for Encrypted Traffic Classification and Analysis," *Int. J. Netw. Manag.*, vol. 25, no. 5, pp. 355–374, 2015.

[8] H. Jiang, A. W. Moore, Z. Ge, S. Jin, and J. Wang, "Lightweight Application Classification for Network Management," in *Proceedings of the IMC*, 2007, pp. 299–304.

[9] D. Tammaro, S. Valenti, D. Rossi, and A. Pescapé, "Exploiting Packet-Sampling Measurements for Traffic Characterization and Classification," *Int. J. Netw. Manag.*, vol. 22, no. 6, pp. 451–476, 2012.

[10] D. Plonka and P. Barford, "Flexible Traffic and Host Profiling via DNS Rendezvous," in *Proceedings of the SATIN*, 2011, pp. 1–8.

[11] A. Tongaonkar, R. Torres, M. Iliofotou, R. Keralapura, and A. Nucci, "Towards Self Adaptive Network Traffic Classification," *Comput. Commun.*, vol. 56, no. 1, pp. 35–46, 2015.

[12] P. Foremski, C. Callegari, and M. Pagano, "DNS-Class: Immediate Classification of IP Flows using DNS," *Int. J. Netw. Manag.*, vol. 24, no. 4, pp. 272–288, 2014.

[13] I. Bermudez, M. Mellia, M. M. Munafò, R. Keralapura, and A. Nucci, "DNS to the Rescue: Discerning Content and Services in a Tangled Web," in *Proceedings of the IMC*, 2012, pp. 413–426.

[14] T. Mori, T. Inoue, A. Shimoda, K. Sato, K. Ishibashi, and S. Goto, "SFMap: Inferring Services over Encrypted Web Flows Using Dynamical Domain Name Graphs," in *Proceedings of the TMA*, 2015, pp. 126–139.

[15] A. Finamore, M. Mellia, M. Meo, M. M. Munafò, and D. Rossi, "Experiences of Internet Traffic Monitoring with Tstat," *IEEE Netw.*, vol. 25, no. 3, pp. 8–14, 2011.

[16] R. Hofstede, P. Čeleda, B. Trammell, I. Drago, R. Sadre, A. Sperotto, and A. Pras, "Flow Monitoring Explained: From Packet Capture to Data Analysis with NetFlow and IPFIX," *Commun. Surveys Tuts.*, vol. 16, no. 4, pp. 2037–2064, 2014.

[17] E. Bocchi, I. Drago, and M. Mellia, "Personal Cloud Storage Benchmarks and Comparison," *IEEE Trans. Cloud Comput.*, vol. PP, no. 99, pp. 1–14, 2015.

[18] T. Callahan, M. Allman, and M. Rabinovich, "On Modern DNS Behavior and Properties," *SIGCOMM Comput. Commun. Rev.*, vol. 43, no. 3, pp. 7–15, 2013.