CNAME Cloaking-based Tracking on the Web: Characterization, Detection, and Protection

Ha Dao Johan Mazel Kensuke Fukuda

***Abstract*—Third-party tracking on the web has been used for collecting and correlating user's browsing behavior. Due to the increasing use of ad-blocking and third-party tracking protections, tracking providers introduced a new technique called CNAME cloaking. It misleads web browsers into believing that a request for a subdomain of the visited website originates from this particular website, while this subdomain uses a CNAME to resolve to a tracking-related third-party domain. This technique thus circumvents the third-party targeting privacy protections.**

The goals of this paper are to characterize, detect, and protect the end-user against CNAME cloaking based tracking. Firstly, we characterize CNAME cloaking-based tracking by crawling top pages of the Alexa Top 300,000 sites and analyzing the usage of CNAME cloaking with CNAME blocklist, including websites and tracking providers using this technique to track users' activities. We also point out that browsers and privacy protection extensions are largely ineffective to deal with CNAME cloaking-based tracking except for Firefox with a developer's version of the uBlock Origin extension. Secondly, we propose a supervised machine learning-based approach to detect CNAME cloaking-based tracking without the on-demand DNS lookup. We show that the proposed approach outperforms well-known tracking filter lists. Finally, to circumvent the lack of DNS API in Chrome-based browsers, we design and implement a prototype of the supervised machine learning-based browser extension to detect and filter out CNAME cloaking tracking, called *CNAMETracking Uncloaker*. Our evaluation shows that *CNAMETracking Uncloaker* is able to filter out CNAME cloaking-based tracking requests without performance degradation when compared with the vanilla setting on the Chrome browser.

***Index Terms*—Privacy, CNAME cloaking-based tracking, Third-party tracking, Machine learning techniques, Countermeasure, Browser extension**

I. Introduction

Web tracking is becoming more and more ubiquitous, thus this brings an increase in privacy concerns from Internet users. In a TRUSTe study, 92% of British Internet users concern their online privacy [3]. A website (first-party domain) has many links to other resources (third-party domains). Some third-party domains are used for user tracking (third-party tracking) to provide functionalities, such as advertising and analytics on the web [4]. For instance, with third-party tracking, advertisements on a website can be customized based on end-users' visits to other websites, which can be frightful for privacy-sensitive users. There are some existing approaches to detect third-party tracking. Many privacy protections take blacklist approaches to detect third-party trackers [5]–[8]. Some works identify tracking requests using cookies [9], or fingerprinting [10], [11]. Other studies intend to detect third-party tracker automatically using machine learning [12]–[14]. These are effective against third-party tracking.

However, web tracking is becoming more and more intricate. One of the emerging techniques is the use of Canonical Name Record or Alias (CNAME) record in Domain Name System (DNS) to hide usual tracking domains that are blocked by browser filter lists and extensions. For instance, website *example.com* embeds a first-party *a.example.com*, which points to a tracking provider *tracker.com* via the CNAME *x.tracker.com*. For instance, website *example.com* embeds a first-party request by subdomain *a.example.com*, which points to a tracking provider *tracker.com* via a CNAME *x.tracker.com*. Because this request is in the first-party context, countermeasures that aim to block third-party tracking are effectively circumvented.

There are some existing methods to detect CNAME cloaking-based tracking. Some network-based blocking methods work at the DNS level, such as NextDNS [15], AdGuard DNS [16], Pi-hole [17] that use to get rid of online tracking. Furthermore, EasyPrivacy [18], AdGuard tracking protection [19], and other filter lists manually add new first-party subdomains which are fronts for CNAME cloaking to these blocklists. However, this approach will dramatically increase the size of the blocklists and these subdomains need to be updated frequently. Besides that, uBlock Origin since version 1.24.1b0 performs a DNS lookup of the hostname loading a resource to determine if the underlying subdomain is related to CNAME cloaking or not. Nevertheless, only Firefox allows uBlock Origin to block CNAME cloaking because the other browsers do not support DNS resolution API [8].

In this paper, we provide a first in-depth analysis of CNAME cloaking-based tracking, propose a supervised machine learning-based method for the detection, and implement *CNAMETracking Uncloaker* browser extension as a countermeasure. The main contributions of the paper are as follows. (1) We first characterize CNAME cloaking-based tracking in Alexa Top 300K sites. We detect 1,739 websites (0.58%) containing CNAME cloaking-based tracking in Alexa 300K sites as of January 2020 by matching with CNAME tracking filter lists (§ IV-C); Those websites are spread across many countries and categories. They use 24 tracking providers in total, and the most common one is Adobe (§ IV-D); By analyzing longitudinal snapshot crawled data of Alexa Top 100K sites (§ IV-E), we show that the usage of CNAME cloaking-based tracking steadily increases from 2016 to 2020;

H. Dao is with Graduate University for Advanced Studies (SOKENDAI), Tokyo, Japan. email: hadao@nii.ac.jp.

J. Mazel is with French National Cybersecurity Agency (ANSSI), France. email: johan.mazel@ssi.gov.fr

K. Fukuda is with National Institute of Informatics and Graduate University for Advanced Studies (SOKENDAI), Tokyo, Japan. email: kensuke@nii.ac.jp.

Manuscript received September, 30, 2020; Revised January, 26, 2021.

[0]This paper is an extended version of work published in Refs. [1], [2].

We then conduct further experiments to investigate the impact of giving consent to CNAME cloaking sites and confirm that there are no significant differences compared to the usage of this phenomena before the user consent is obtained (§ IV-F) as a new contribution. We also evaluate the detection ability of such tracking for major browsers and extensions (§ V). (2) Next, we propose the supervised machine learning-based method to detect CNAME cloaking-based tracking without the on-demand DNS lookup (§ VI); Through the comprehensive analysis, we demonstrate the effectiveness of our method. (3) As a new contribution beyond Refs. [1], [2], we design and implement a prototype browser extension of the supervised machine learning approach to protect the end-user against CNAME cloaking-based tracking, named *CNAMETracking Uncloaker* (§ VII). The current best counter-measure strongly depends on realtime name resolution (only supported by Firefox browser), but our extension intends to distinguish requests using CNAME cloaking-based tracking in Chrome-based browsers. Our experiment shows that the performance overhead is acceptable when compared with the vanilla setting on the Chrome browser.

II. Background and Term Definitions

A. Background

1) Third-party web tracking: Privacy leakage occurs through communications with trackers. Third-party web tracking refers to the practice by which an entity (the tracker), other than the website directly visited by the user, identifies and collects information about web users.

From the view of website administrators, user tracking is useful for a variety of purposes such as behavioral advertising or website analytics. On the user's side, the larger number of browsing profiles, the greater loss of privacy.

2) Privacy protection techniques: Several privacy protection techniques have been designed to protect end-users from third-party tracking, including network-based blocking, extensions, and browser itself.

Network-based blocking methods use address-based blacklists in order to block access to certain domains (DNS blocking) and modify web traffic (interception proxies), which work independently of the underlying application or browser [20].

Some anti-tracking extensions work effectively to detect third-party tracking, such as Ghostery [21], Disconnect [7], and uBlock Origin [8]. Some browsers also have built-in privacy protection features to protect end-users from third-party tracking, such as Firefox [22], Brave [23], and Tor Browser [24]. Firefox introduces Enhanced Tracking Protection (ETP) feature from Firefox version 69. It blocks user profile from browsing behavior observation across websites [25]. Brave has a feature called Shields which protects user's privacy by blocking ads and trackers, cookies, malicious code, and malicious sites [23]. The Tor Browser is a browser based on the onion routing tool Tor and Mozilla's Extended Support Release (ESR) Firefox branch to enhance privacy and security. It includes both HTTPS-Everywhere and NoScript extensions which respectively enable HTTPS when possible, and allow users to block JavaScript [26].

B. Term definitions

We first define some terms we use throughout this paper:

1) Subdomain: In the DNS hierarchy, a subdomain is any domain, which is underneath a main domain. Subdomains are used to organize or divide contents of a website into specific sections. For example, *a.example.com* and *b.example.com* are subdomains of domain *example.com*.

2) CNAME cloaking-based tracking: The usage of DNS CNAME records coupled with Content Delivery Network (CDN) is increasingly commonplace to improve website load times, reduce bandwidth costs, and increase content availability and redundancy.

CNAME has also been used for user tracking. Tracking providers ask their clients to delegate a subdomain for data collection and tracking and link it to an external server using a CNAME DNS record [27]. This technique, called *CNAME cloaking-based tracking*, uses CNAME to disguise requests to a third-party tracker as first-party ones. We also define an HTTP request by this subdomain is a *request linked to CNAME cloaking-related tracking.*

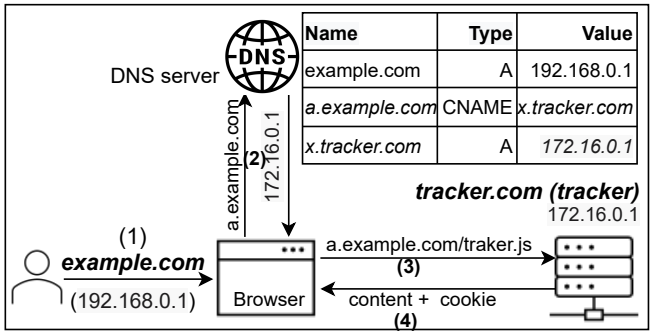

Fig. 1. The process of the browser connecting to tracking provider by CNAME cloaking-based tracking

Figure 1 shows the process of the browser connecting to a third-party tracking server by CNAME cloaking-based tracking to setup third-party cookies in the first-party context:

1) An end-user types the URL of website *example.com* (192.168.0.1) into his/her browser and presses return. This website embeds a subdomain *a.example.com*
2) The browser looks up *a.example.com* on the DNS server and finds an IP address 172.16.0.1 of tracking provider *tracker.com*.
3) Browser connects to the tracking provider web server *tracker.com* and asks for request script *tracker.js*.
4) Server *tracker.com* sends over the requested content along with cookies information. Browser accepts the content and these persistent cookies, which are stored under the domain name *example.com*. From there, the tracking provider *tracker.com* thus tracks activities of this end-users on the website *example.com*.

CNAME cloaking-based tracking circumvents the third-party targeting privacy protections but it has a limitation. These persistent first-party subdomain-related cookies make it more difficult for third-parties to track users across websites by removing the simple mapping of each user to a single cookie linked to a single (third-party) domain.

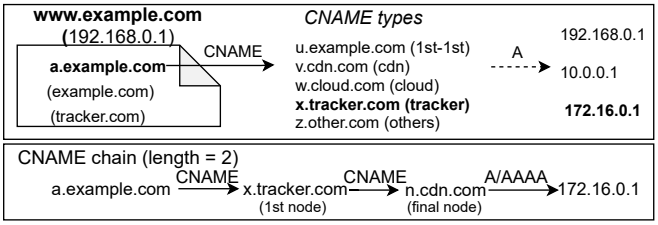

Fig. 2. Overview of CNAME chain.

TABLE I
SUMMARY OF CRAWLED DATA IN ALEXA TOP 300K SITES (JAN 2020).

Metrics			Numbers	Percentage
3rd party requests			14,640,568	54,27%
1st party requests	domain		5,919,965	21.94%
	subdomain	w/o CNAME	3,245,361	12.03%
		w/ CNAME	**3,172,304**	**11.76%**
Total requests			26,978,198	100%

C. CNAME chain

CNAME chain corresponds to a series of CNAMEs from the initial first-party subdomain to all CNAME nodes before the resolution to an IP address (see Figure 2). We consider four CNAME types for a CNAME chain:

1) First-party type: The domain of the final node in a CNAME chain is the same as the domain of the considered HTTP request, or the IP addresses of both the final node and the second-level domain are the same (*u.example.com*).
2) CDN type: The domain of nodes in a CNAME chain is used for CDN service (*v.cdn.com*).
3) Cloud and other types: The domain of nodes in a CNAME chain is used for other activities, such as cloud storage or firewall (*w.cloud.com*, *z.other.com*).
4) Tracker type: The domain of nodes in a CNAME chain is used for tracking user activities (*x.tracker.com*).

III. DATA COLLECTION AND BLOCKLIST-BASED CNAME CLOAKING-BASED TRACKING DETECTION

In this section, we describe the data collections and explain our methodology to detect CNAME cloaking-based tracking with blocklists.

A. Websites selection and Data collection

The first step is the selection of websites that would be most appropriate for our work. We use the popularity index from Alexa [28] in all of our measurements, similar to past literature [11], [20], [29]. To characterize CNAME cloaking-based tracking, we use OpenWPM [11] to conduct large-scale automatic crawls on Alexa Top 300K sites. OpenWPM is based on Firefox version 52 and allows collecting all the HTTP/HTTPS requests emitted and their responses for each site. We performed the crawls with default settings in January 2020, with three IP addresses in Japan (Table I).

In addition, in order to track the longitudinal behavior of CNAME cloaking-based tracking, we also rely on four other datasets (see Table II). We collected two datasets on Alexa Top 100K sites with OpenWPM in April 2018 and January 2020. The other two datasets are publicly available in Princeton Web Census Data [11]. They were collected in January 2016 and February 2017 and targeted Alexa Top 100K sites. These datasets were also crawled with OpenWPM, so all the data sources are compatible and comparable. Note that the contents of Alexa lists are not the same among these four datasets because Alexa lists themselves are updated daily and change significantly from one day to the next [30]. The list used for each crawl is described in "List gen." column of Table II.

TABLE II
LONGITUDINAL SNAPSHOT DATASETS.

Time	Alexa	List gen.	Requests	Firefox version
Jan 2016	100K	01/2016	9,487,367	41
Feb 2017	100K	11/2016	10,964,374	45
Apr 2018	100K	03/2018	9,926,080	52
Jan 2020	100K	12/2019	9,647,506	52

Furthermore, the instability of the Alexa Top list drastically increased in January 2018 [30]. So, in order to make a fair comparison, we also use the intersection (26,162 sites) of the four Alexa Top 100k sites above.

Note that, we found a publicly available HTTP Archive (HAR) dataset that provides historical data to quantitatively illustrate how the web is evolving [31]. However, we recognized two limitations of this dataset. First, The HAR dataset is periodically crawled the top websites that come from the Chrome User Experience Report, but there is no ranking value in this dataset to assess whether end-users are actually impacted by CNAME cloaking. Second, there is no way to control its crawling and publishing schedule to obtain the up-to-date DNS data. Due to these reasons, we decided to use the dataset as described above for our measurement.

B. Blocklist-based CNAME cloaking detection

1) CNAME lookup: First of all, we separate the generic Top-Level Domain (gTLD) and country-code top-level domain (ccTLD) from the visited website for all HTTP requests using the Public Suffix List [32]. We only keep subdomain of an HTTP request if it is not null and its second-level domain is the same as the visited website domain. We look up and check CNAME records for each subdomain. We then resolve each CNAME answer set by DNS. We save all nodes in CNAME chain (see § II-C) to analyze the CNAME cloaking behind first-party requests. We find that 45.73% of the HTTP requests are first-party requests in 2020 (Table I). We then only keep 11.76% of the HTTP requests that contain first-party CNAME.

Looking up CNAMEs for the longitudinal data, we additionally check historical forward DNS (FDNS) datasets provided by Rapid7 [33]. The coverage of the FDNS data in our CNAME data is not perfect. It missed 10% of CNAMEs in 2018 and 30% in 2016 and 2017. We intend to use DNSDB [34] in future research to improve this coverage.

2) CNAME cloaking-based tracking detection with blocklists: To detect CNAME cloaking-based tracking, we use an approach based on wildcards matching of tracking filter list.

First, we discard CNAME-related subdomains that are categorized as first-party type. We classify a CNAME chain as *first-party* if the domain of the final node in this chain is the same as the domain of the considered HTTP request, or if the IP addresses of both the final node and the second-level domain are the same.

We then intend to detect CNAME cloaking-based tracking inside the remaining subdomains. We apply wildcard matching based on well-known tracking filter lists: EasyPrivacy list [35] and AdGuard tracking protection filter [19]. EasyPrivacy list consists of nine sublists and the Adguard tracking filter list consists of eleven sublists. They contain many rules that remove all forms of tracking, including web bugs, tracking scripts, and information collectors, thereby protecting user personal data. Focusing on tracking domains, we select the *third-party tracking domains*, the *international third-party tracking domains*, the third-party domain from *third-party tracking services*, and the third-party domain from *International third-party tracking services* sublists from EasyPrivacy list and the *tracking servers list* sublist from AdGuard tracking protection filter as of February 5, 2020. These blocklists are partly overlapping. We build the union of the two filter lists above to make a *CNAME tracking filter list*. Then, we build regular expressions from tracking domains to match with CNAME behind all remaining subdomains. For example, *eulerian.net∧third-party* is changed to *.eulerian.net.$*. This rule matches any CNAME ending with *.eulerian.net.*; We can thus detect any CNAME cloaking-based tracking from tracking provider Eulerian [36]. Finally, we inspect individual CNAME nodes in all CNAME chains using this customized filter list. If any node in a CNAME chain is flagged by this list, we classify this CNAME chain as a potential *tracker* that flag by 62 domains from *our CNAME tracking filter list*.

To avoid false positives, we then group these CNAME chains by domain and inspect them manually one by one. We first validate them by observing the activities which store uniquely cookie in the browser under visited domain name. We also gather information about these domains to identify whether they belong to any tracking provider. Using this analysis, we finally consider 28 domains are used for CNAME cloaking-based tracking and flag these chains as *tracker*.

We furthermore use CDN lists [37], [38] to check if remaining CNAME chains are *CDN*. If it is not the case, we consider them as *Others* (see Figure 2).

IV. Characterizing CNAME cloaking-based tracking

A. CNAME cloaking-based tracking analysis

Having gathered the CNAME chains using CNAME cloaking-based tracking, we concentrate on analyzing websites and tracking providers linked to CNAME cloaking-based tracking.

We consider the ranking, the country, and the category of websites containing CNAME cloaking-based tracking. For the website ranking, we assess how a real user would be affected in the real world by this type of tracking by examining the Empirical Cumulative Distribution Function (ECDF) of these websites. For the website country using CNAME cloaking-based tracking, we analyze them based on top-level domain (TLD), Whois information, and IP Geolocation. First of all, if the TLD of a website corresponds to a country (i.e., ccTLD), we attribute that website to the country. By doing that, we identify the country of 94,560 websites. Then, for international TLDs, we use Whois information to determine 171,370 websites country. Finally, for 34,070 remaining domains, we use the IP Geolocation to determine the country. We are aware that, if a website uses cloud-based security, proxy, or DNS-based service, then the geolocation of returned IP address could be unreliable for our purpose. However, this error was negligible, especially, there are a small number of such CNAME cloaking websites as shown in a later section (see § IV-C). In addition, IP Geolocation sometimes returns incorrect results [39]. To overcome this limitation, we make a majority voting via ip-api.com [40], freegeoip.app [41], and MaxMind [42] to give more robustness to the Geolocation assignment. In the 1,307 cases of three databases return difference results or return null, we set these websites to an unknown country. For the website category, we use FortiGuard Web Filtering [43] dataset from January 2020 for the website category classification.

Finally, we consider tracking providers behind CNAME cloaking-based tracking by linking 28 domains are used for CNAME cloaking to 24 tracking providers using Disconnect's blocklist [44].

TABLE III
CNAME types of first-party request by subdomain (Alexa 300K sites in 2020).

Metric	1st-1st	Tracker	CDN, Cloud and others
HTTP requests	1,839,728/57.99%	**3,484/0.11%**	1,329,092/41.90%
Subdomains	48,365/39.47%	**1,803/1.47%**	72,376/59.06%

B. CNAME chains structure

In this section, we focus on the characteristics of CNAME chains for first-party subdomain in Alexa Top 300K sites. Firstly, we present the CNAME usage of first-party request by subdomain in Table III. The most common CNAME type is requests referring to resources of the first party (57.99%). CDN and cloud also represent a large proportion of CNAME type (41.90%). Overall, we detect 3,484 CNAME cloaking-based tracking URLs. Furthermore, we find that these URLs belong to 1,739 websites (0.58%) on Alexa Top 300K sites.

Then, we breakdown the number of nodes in CNAME chains for first-party subdomains in our latest dataset (Alexa Top 300K sites in 2020) in Figure 3. We observe that about 80% of CNAME chains are very simple, just consisting of one CNAME. However, we also observe longer chains whose maximum length is six. These longest chains are mainly used by Microsoft likely for load balancing. This result suggests that checking only the first CNAME might be not enough for detecting CNAME cloaking-based tracking, because tracker websites may appear in intermediate nodes in the chain.

Finally, we show the breakdown of CNAME types regarding their position in CNAME chains in Figure 4. We note that the position represents the location of a CNAME in a CNAME

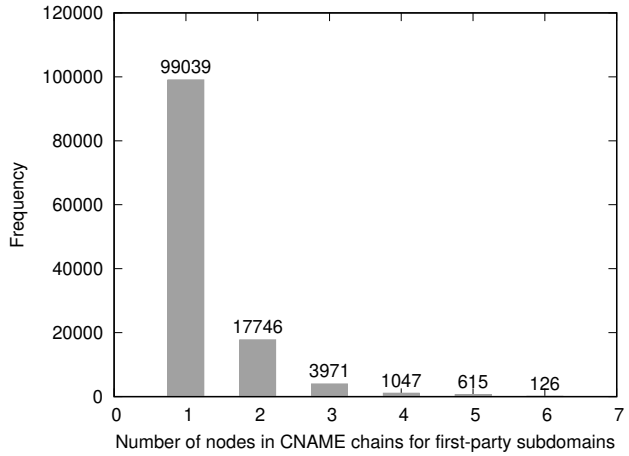

Fig. 3. The number of nodes in CNAME chains for first-party subdomains (Alexa Top 300K sites in 2020).

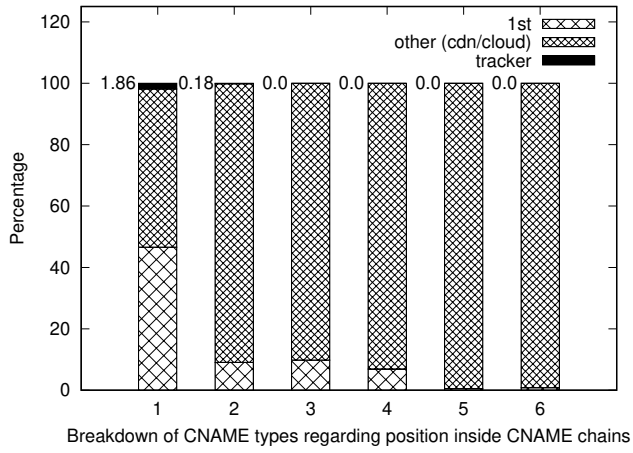

Fig. 4. Breakdown of CNAME types regarding position inside CNAME chains (Alexa Top 300K sites in 2020).

chain. For example, the first position of a CNAME chain with two nodes *x.tracker.com* and *n.cdn.com* is the CNAME *x.tracker.com* (see Figure 2). In Alexa Top 300K sites, the tracking-related domain inside a CNAME chain is mainly located at the first position. We however also observe some tracking domains in the second position.

C. Websites using CNAME cloaking-based tracking

Next, we focus on the characteristics of websites containing CNAME cloaking-based tracking.

Figure 5 presents the Empirical Cumulative Distribution Function (ECDF) of the Alexa ranking of websites containing CNAME cloaking-based tracking. These websites are spread across the Alexa ranking. It illustrates that 30% of the CNAME cloaking-based tracking belongs to the top 20K websites. Popular websites use more CNAME cloaking-based tracking.

Then, we discuss the website category of websites containing CNAME cloaking-based tracking that shown in Figure 6. For 1,739 websites containing CNAME cloaking, the percentages of websites in Business, Information Technology, Shopping and Finance are 22.0%, 17.3%, 11.8%, and 9.7%, respectively. In addition, for the proportion of website using CNAME cloaking inside each category, the percentages of these websites account for 0.6%, 0.6%, 1.2%, and 2.4%, respectively. Overall, various website categories use CNAME cloaking.

Next, we analyze the website country of websites containing CNAME cloaking-based tracking[1] that shown in Figure 7. We observe that 55.1% of websites are located in the United States, 5.6% are located in Germany, 5.3% are located in the United Kingdom, 5.0% are located in Japan, 4.9% are located in Canada, and other countries have significantly lower percentages. In addition, for the proportion of website using CNAME cloaking inside each country, the percentage of the United States, Germany, the United Kingdom, Japan, and Canada are 1.1%, 1.7%, 1.2%, 1.0%, and 0.7%, respectively. Overall, there is not a big difference among website using CNAME cloaking regarding country.

In summary, we intended to investigate any biases, but we do not observe significant biases regarding website categories and website countries of sites containing CNAME cloaking-based tracking. In contrast, websites using this tracking technique are widely spread in many countries and categories.

D. Tracking providers using CNAME cloaking-based tracking

We provide the breakdown of tracking providers behind CNAME cloaking-based tracking in Figure 8. We confirm 56 tracking providers using this technique. The major player in Alexa Top 300K sites is Adobe (52.5%). Besides Adobe, we see some well-known tracking providers, such as Pardot [45], Act-on [46], Oracle [47], and Webtrekk [48] (25.7%, 6.3%, 3.0%, and 2.5%, respectively).

Moreover, Table IV shows the breakdown of tracking providers inclusion in website by website category. We observe that tracking providers were distributed in different types of websites. Except Travel category (92% for Intent), Adobe is the most popular tracking provider in almost all categories. The second one is Pardot. Furthermore, Table V shows the breakdown of the tracking providers inclusion in website by website country. Tracking providers cooperating with websites such as Act-on, PostafiliatePro, Pardot, Adobe, and Oracle are mainly located in the United States (80.2%, 66.7%,61.5%, 56.7%, and 52.8%, respectively). We also observe that some tracking providers are mainly located in specific countries, e.g., Eulerian in France (54.8%) and Webtrekk in Germany (68.9%). Again, Adobe and Pardot are the most popular tracking providers in almost all countries, except France (Eulerian with 32.9%).

Finally, we further investigate the number of tracking providers on each website. Most websites (1,707) deploy only one tracking provider, as expected. However, we also find 31 websites using two providers, and one website *mytoys.de* using three providers (Webtrekk, Otto Group, and Adclear). Typical

[1] The website country for 1,739 sites containing CNAME cloaking-based tracking is determined by ccTLD (434 sites; 24.96%), Whois (1,054 sites; 60.61%), and IP Geolocation (251 sites; 14.43%). The websites detected by the IP Geolocation are identified as the United States and Canada (222 sites), European countries (22 sites), and others (seven sites). We manually confirm that most results are not affected by CDN.

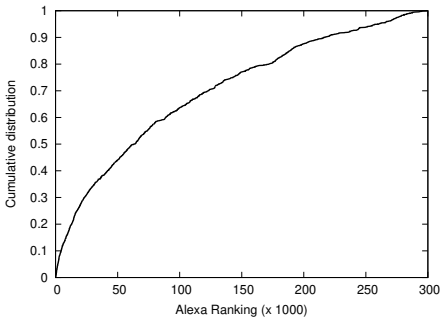

Fig. 5. ECDF of the Alexa ranking of websites containing CNAME cloaking-based tracking (Alexa Top 300K sites in 2020).

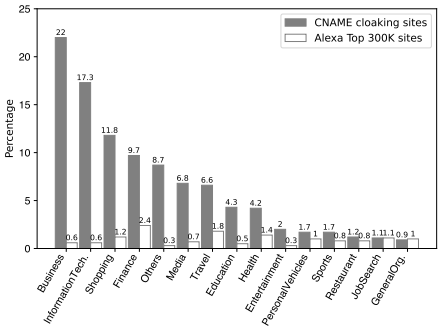

Fig. 6. Breakdown of websites containing CNAME cloaking-based tracking by website category (Alexa Top 300K sites in 2020).

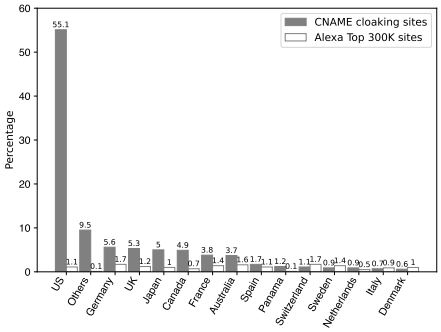

Fig. 7. Breakdown of websites containing CNAME cloaking-based tracking website country (Alexa Top 300K sites in 2020).

TABLE IV

BREAKDOWN OF TRACKING PROVIDERS INCLUSION IN WEBSITE BY WEBSITE CATEGORY. THE VALUES HAVE THE FOLLOWING MEANING: RAW/PERCENTAGE FOR CATEGORY/PERCENTAGE FOR TRACKING PROVIDER. THE SIGNIFICANT PERCENTAGES (>20%) ARE SHOWN IN BOLD.

Category	Adobe	Pardot	Act-On	Oracle	Webtrekk	Eulerian	Segment	Intent	PostafiliatePro	Others	Total
Business	141/15.1/**36.5**	157/**34.5**/**40.7**	39/**35.1**/10.1	13/**24.5**/3.4	14/**31.1**/3.6	5/11.9/1.3	7/**24.1**/1.8	0/0/0	2/16.7/0.5	8/11.6/2.1	**386**/NA/100
Information Technology	95/10.2/**30.4**	139/**30.5**/**44.6**	25/**22.5**/8.0	16/**30.2**/5.1	6/13.3/1.9	7/16.7/2.2	8/**27.6**/2.6	0/0/0	4/**33.3**/1.3	12/17.4/3.8	**312**/NA/100
Shopping	157/16.9/**73.0**	5/1.1/2.3	1/0.9/0.5	3/5.7/1.4	10/**22.2**/4.7	10/**23.8**/4.7	4/13.8/1.9	0/0/0	2/16.7/0.9	23/**33.3**/10.7	**215**/NA/100
Finance	117/12.6/**68.8**	24/5.3/14.1	7/6.3/4.1	6/11.3/3.5	4/8.9/2.4	3/7.1/1.8	1/3.4/0.6	1/4.0/0.6	0/0/0	7/10.1/4.1	**170**/NA/100
Media	96/10.3/**80.0**	5/1.1/4.2	2/1.8/1.7	0/0/0	5/11.1/4.2	3/7.1/2.5	1/3.4/0.8	1/4.0/0.8	0/0/0	7/10.1/5.8	**120**/NA/100
Travel	64/6.9/**54.2**	10/2.2/8.5	6/5.4/5.1	0/0/0	2/4.4/1.7	8/19.0/6.8	0/0/0	23/**92.0**/19.5	0/0/0	5/7.2/4.2	**118**/NA/100
Education	18/1.9/**24.0**	41/9.0/**54.7**	8/7.2/10.7	4/7.5/5.3	0/0/0	0/0/0	4/13.8/5.3	0/0/0	0/0/0	0/0/0	**75**/NA/100
Health	45/4.8/**61.6**	19/4.2/**26**	3/2.7/4.1	2/3.8/2.7	1/2.2/1.4	0/0/0	1/3.4/1.4	0/0/0	2/16.7/2.7	0/0/0	**73**/NA/100
Entertainment	31/3.3/**88.6**	1/0.2/2.9	2/1.8/5.7	0/0/0	0/0/0	1/2.4/2.9	0/0/0	0/0/0	0/0/0	0/0/0	**35**/NA/100
Personal Vehicles	24/2.6/**80.0**	2/0.4/6.7	0/0/0	2/3.8/6.7	0/0/0	0/0/0	0/0/0	0/0/0	0/0/0	2/2.9/6.7	**30**/NA/100
Sports	20/2.1/**69.0**	4/0.9/13.8	1/0.9/3.4	1/1.9/3.4	0/0/0	1/2.4/3.4	2/6.9/6.9	0/0/0	0/0/0	0/0/0	**29**/NA/100
Restaurant	16/1.7/**80.0**	3/0.7/15.0	0/0/0	0/0/0	0/0/0	1/2.4/5.0	0/0/0	0/0/0	0/0/0	0/0/0	**20**/NA/100
Job Search	10/1.1/**52.6**	7/1.5/**36.8**	2/1.8/10.5	0/0/0	0/0/0	0/0/0	0/0/0	0/0/0	0/0/0	0/0/0	**19**/NA/100
General Organizations	6/0.6/**35.3**	7/1.5/**41.2**	2/1.8/11.8	0/0/0	0/0/0	0/0/0	0/0/0	0/0/0	0/0/0	2/2.9/11.8	**17**/NA/100
Others	91/9.8/**59.5**	31/6.8/**20.3**	13/11.7/8.5	6/11.3/3.9	3/6.7/2.0	3/7.1/2.0	1/3.4/0.7	0/0/0	2/16.7/1.3	3/4.3/2.0	**153**/NA/100
Total	**931**/100/NA	**455**/100/NA	**111**/100/NA	**53**/100/NA	**45**/100/NA	**42**/100/NA	**29**/100/NA	**25**/100/NA	**12**/100/NA	**69**/100/NA	**1,772**/NA/NA

TABLE V

BREAKDOWN OF TRACKING PROVIDERS INCLUSION IN WEBSITE BY WEBSITE COUNTRY. THE VALUES HAVE THE FOLLOWING MEANING: RAW/PERCENTAGE BY COUNTRY/PERCENTAGE BY TRACKING PROVIDER. THE SIGNIFICANT PERCENTAGES (>20%) ARE SHOWN IN BOLD.

Country	Adobe	Pardot	Act-On	Oracle	Webtrekk	Eulerian	Segment	Intent	PostafiliatePro	Others	Total
United States	528/**56.7**/**54.5**	280/**61.5**/**28.9**	89/**80.2**/9.2	28/**52.8**/2.9	1/2.2/0.1	1/2.4/0.1	11/**37.9**/1.1	5/**20.0**/0.5	8/**66.7**/0.8	18/**26.1**/1.9	**969**/NA/100
Germany	38/4.1/**35.5**	8/1.8/7.5	1/0.9/0.9	1/1.9/0.9	31/**68.9**/**29.0**	1/2.4/0.9	0/0/0	2/8.0/1.9	0/0/0	25/**36.2**/**23.4**	**107**/NA/100
United Kingdom	58/6.2/**62.4**	22/4.8/**23.7**	2/1.8/2.2	2/3.8/2.2	1/2.2/1.1	2/4.8/2.2	1/3.4/1.1	3/12.0/3.2	0/0/0	2/2.9/2.2	**93**/NA/100
Japan	36/3.9/**40.9**	51/11.2/**58.0**	0/0/0	1/1.9/1.1	0/0/0	0/0/0	0/0/0	0/0/0	0/0/0	0/0/0	**88**/NA/100
Canada	50/5.4/**58.1**	22/4.8/**25.6**	4/3.6/4.7	3/5.7/3.5	0/0/0	6/14.3/7	0/0/0	1/4.0/1.2	0/0/0	0/0/0	**86**/NA/100
France	15/1.6/**21.4**	14/3.1/**20.0**	1/0.9/1.4	2/3.8/2.9	0/0/0	23/**54.8**/**32.9**	0/0/0	3/12.0/4.3	0/0/0	12/17.4/17.1	**70**/NA/100
Australia	48/5.2/**71.6**	10/2.2/14.9	2/1.8/3.0	3/5.7/4.5	0/0/0	0/0/0	4/13.8/6.0	0/0/0	0/0/0	0/0/0	**67**/NA/100
Spain	20/2.1/**64.5**	0/0/0	0/0/0	0/0/0	1/2.2/3.2	8/19.0/**25.8**	0/0/0	0/0/0	0/0/0	2/2.9/6.5	**31**/NA/100
Panama	2/0.2/10.0	10/2.2/**50.0**	1/0.9/5.0	2/3.8/10	0/0/0	0/0/0	4/13.8/**20.0**	0/0/0	1/8.3/5	0/0/0	**20**/NA/100
Switzerland	11/1.2/**57.9**	5/1.1/**26.3**	1/0.9/5.3	1/1.9/5.3	0/0/0	0/0/0	0/0/0	0/0/0	0/0/0	1/1.4/5.3	**19**/NA/100
Sweden	7/0.8/**43.8**	6/1.3/**37.5**	2/1.8/12.5	1/1.9/6.3	0/0/0	0/0/0	0/0/0	0/0/0	0/0/0	0/0/0	**16**/NA/100
Netherlands	9/1.0/**60.0**	3/0.7/**20.0**	0/0/0	0/0/0	2/4.4/13.3	0/0/0	0/0/0	0/0/0	0/0/0	1/1.4/6.7	**15**/NA/100
Italy	8/0.9/**61.5**	1/0.2/7.7	0/0/0	1/1.9/7.7	2/4.4/15.4	0/0/0	0/0/0	0/0/0	0/0/0	1/1.4/7.7	**13**/NA/100
Denmark	8/0.9/**80.0**	0/0/0	0/0/0	2/3.8/**20.0**	0/0/0	0/0/0	0/0/0	0/0/0	0/0/0	0/0/0	**10**/NA/100
Others	93/10.0/**55.4**	23/5.1/13.7	8/7.2/4.8	6/11.3/3.6	7/15.6/4.2	1/2.4/0.6	9/**31.0**/5.4	11/**44.0**/6.5	3/**25.0**/1.8	7/10.1/4.2	**168**/NA/100
Total	**931**/100/NA	**455**/100/NA	**111**/100/NA	**53**/100/NA	**45**/100/NA	**42**/100/NA	**29**/100/NA	**25**/100/NA	**12**/100/NA	**69**/100/NA	**1,772**/NA/NA

pairs of the providers are the combination between Adobe and other tracking providers, such as (Adobe and Oracle), (Adobe and Webtrekk), or (Adobe and Pardot). We do not identify any plausible reasons of deploying multiple providers, but they might be used for different purposes (e.g., analytics and advertisement).

We conclude that, besides the biggest player Adobe, CNAME cloaking tracking providers operate on many website categories and countries.

E. Longitudinal analysis of CNAME cloaking-based tracking

In this section, we analyze the longitudinal evolution of the number of websites using CNAME cloaking-based tracking. Figure 9 indicates the number of websites using CNAME cloaking-based tracking in Alexa 100K sites. We combine two crawled datasets and two DNS lookup datasets: (1) for the crawled data, the number of websites in each Alexa 100K and those in the overlap among all Alexa 100K datasets (26,126 sites); (2) for two DNS lookup datasets, DNS lookup in 2020

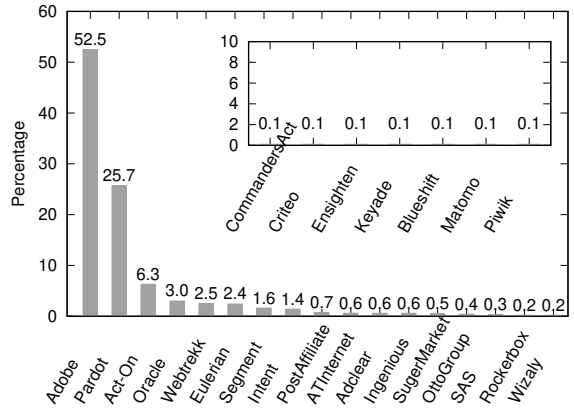

Fig. 8. Tracking providers providing CNAME cloaking-based tracking (Alexa Top 300K sites in 2020).

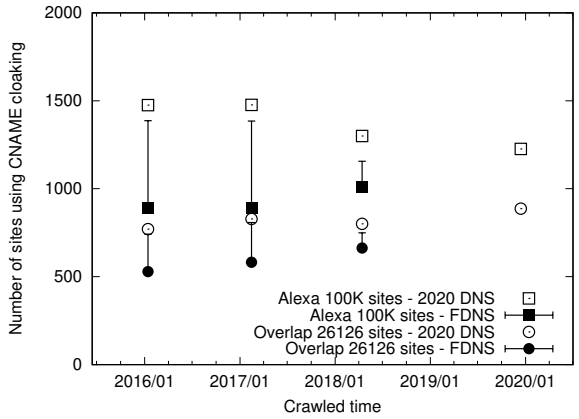

Fig. 9. Websites containing CNAME cloaking-based tracking along time.

and lookup with the FDNS data (collected in February 2017, the oldest available snapshot, and June 2018). We then plot four combinations: the number of websites in each Alexa 100K sites with 2020 DNS (white rectangles) and with FDNS (black rectangles). Those in the overlap among all Alexa 100K datasets with DNS in 2020 (white circles) and with FDNS (black circles). The error bars in the figure show the number of unsolved CNAMEs due to the coverage of the FDNS data.

We discuss the growth of websites introducing CNAME cloaking-based tracking over the years. At a glance, the number of websites containing CNAME cloaking-based tracking is slightly decreasing in Alexa Top 100K websites with the latest DNS (white rectangles). However, this decrease is due to biases of DNS lookup. Considering the historical DNS data (black rectangles), we conjecture the presence of an increasing trend in the use of CNAME cloaking. However, the large number of unsolved CNAMES in 2016 and 2017 (represented by the error bars in the figure) does not allow to confirm this. We see an increasing trend in the overlapping websites (white and black circles) with smaller error bars. Although the unsolved CNAMES in 2016 and 2017 for the yearly Alexa limit the strength of our conclusion, the evolution between 2018 and 2020 for yearly Alexa, and the overall trend in the overlapping websites, allow us to confirm an increasing trend along the observed years.

TABLE VI
DETECTION PERFORMANCE: EASYPRIVACY LIST AND ADGUARD TRACKING PROTECTION FILTER (ALEXA TOP 300K SITES IN 2020).

Metric	AdGuard Tracking	EasyPrivacy	All (combined)
HTTP requests	1,433/41.13%	2,707/77.70%	2,713/77.87%
Subdomains	444/24.63%	1,313/72.82%	1,316/72.99%
Sites	422/24.27%	1,262/72.57%	1,265/72.74%

F. Impact of giving consent to CNAME cloaking sites

We also conduct extra experiments to evaluate the impact of providing consent as legally required by General Data Protection Regulation (GDPR) [49] to websites with CNAME cloaking. We pickup 1000 sites in Alexa top 300K sites (top 500 sites, randomly middle 250 sites, and randomly bottom 250 sites). On a clean browser session, we load the website. If there is no cookie notification or only a text simply informing the users about the site's use of cookie, we stop there. We find 917 sites (91.7%) as *no banner* (the publishers do not inform the end-user of data collection) and 19 sites (0.19%) as *notification only* (the notification simply informing the users about the site's use of cookies) category. For 64 (0.64%) remaining sites, we crawl them twice. In the first time, we save all requests and responses in these sites without human manipulation and find 31 sites (0.31%) as *Accepted only* (the notification does not offer a way to refuse consent) and 33 sites (0.33%) as *More options* (the user can make their choice in the cookies notification by clicking *accept*, *reject*, or *more setting*) category. In the second time, we click to accept consent on the banner, record the requests (if any). Comparing the difference between *Accepted only* and *More options* categories in the two crawls, we confirm that there are four websites already embed a subdomain-related request to hidden CNAME cloaking based-tracking without obtain user consent. These results demonstrate that there is no significant effect by giving consent to the CNAME cloaking sites in our measurement.

V. MEASURING THE EFFECTIVENESS OF THE CURRENT IN-BROWSER PROTECTION TECHNIQUES AGAINST CNAME CLOAKING

We analyze and compare browsers and extensions regarding privacy protection against CNAME cloaking-based tracking.

A. Filter list

In order to block CNAME cloaking-based tracking, EasyPrivacy [35] and AdGuard tracking protection [19] require the identification of first-party subdomains which are fronts for CNAME cloaking. They follow the Adblock Plus filter syntax. For example, EasyPrivacy has a rule to block tracking provider Eulerian: *f7ds.liberation.fr^*. So, when website *liberation.fr* makes a request to the third-party tracker Eulerian through *f7ds.liberation.fr*, the request is blocked.

We assess the efficiency of these filter lists as countermeasures. We use *Adblockparser* [50] that can parse Adblock Plus filters to directly match blocking list rules with all HTTP requests in the Alexa 1,739 sites that contain CNAME

cloaking-based tracking in Table III. Note that, *Adblockparser* has some limitations [51], but it does not impact our measurement for request-related to CNAME cloaking.

We inspect individual CNAME cloaking-based tracking URLs using these well-known filter lists in January 2020. The results of this experiment are shown in Table VI. We find that 2,713 CNAME cloaking-based tracking URLs have been flagged by these filter lists. This represents 77.87% of all CNAME cloaking-based tracking URLs in Alexa Top 300K sites. Besides that, the EasyPrivacy list detects almost as much CNAME cloaking-based tracking as combined lists. This means that CNAME cloaking domains detected by Adguard tracking filter list are almost always detected by EasyPrivacy. Overall, tracker blocking lists thus do not effectively deal with CNAME cloaking-based tracking. Subdomains being used for CNAME cloaking may change often, which makes day-to-day filter lists updating tedious and time-consuming, and thus explain filter list poor performances.

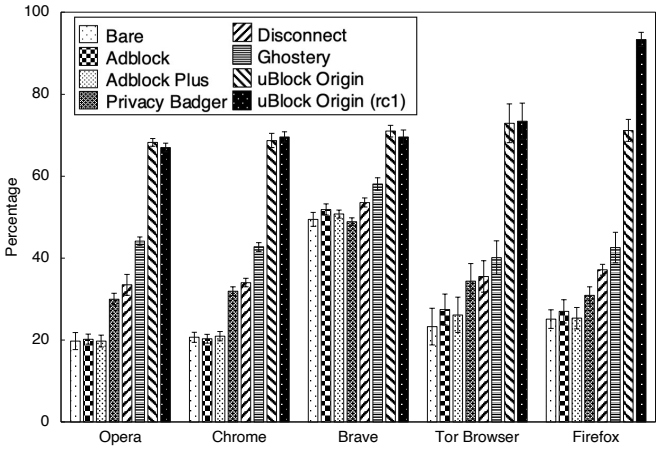

Fig. 10. Detection performance of browsers and extensions regarding websites containing CNAME cloaking-based tracking. The mean and standard deviation are computed on three crawls.

B. Browsers and extensions

Some browsers focus on security and privacy by blocking trackers. Browser extensions also use several techniques (such as blocklisting, or traffic monitoring) to block third-party tracking. We evaluate the ability of common browsers and extensions to block CNAME cloaking-based tracking.

We investigate five major browsers and six popular privacy protecting extensions that support these browsers. We choose following popular browsers [52]: Chrome 80.0 [53], Opera 66.0 [54], Brave 1.4.92 [23], Firefox 73.0 [22] and Tor Browser 9.0.2 [24]. Regarding extensions, we use two criteria: blocking trackers and supporting multiple browsers. The privacy extensions that meet our criteria are Adblock 4.5.0 [5], Adblock Plus 3.7 [6], Privacy Badger 2020.1.13 [18], Disconnect 5.19.3 [7], Ghostery 8.4.6 [21], uBlock Origin 1.24.4 [8] and 1.24.5rc1 (developer's version) [55]. Ublock Origin 1.24.5rc1 has an anti CNAME cloaking-based tracking feature [55]. We include this version to provide an up-to-date picture of CNAME cloaking-based tracking counter-measures. We then collect all the HTTP requests and responses on the 1,739 websites containing CNAME cloaking-based tracking in Table III. We use Atrica[2] [56], a multi-browser crawling library, to gather data on websites with CNAME cloaking-based tracking. To conduct a general comparison of browsers and privacy protection techniques, we crawl 1,739 websites using 40 different profile configurations (five browsers × eight extensions including the vanilla/bare setting). All the measurements were performed in March 2020 with three IP addresses in Japan. One crawling took approximately 4 to 6 hours on commodity hardware.

To reduce measurement error, we conducted three crawls and computed the relative standard error of the mean percentage of websites using CNAME cloaking-based tracking. We notice that there are also several possible sources of noise in our data. Some of these are internal and known, such as failure to connect to a website on a special time, or may also be external factors, such as network unreliability. To make a fair comparison, we set the website crawl timeout to 60 seconds. After this duration, if any website does not finish loading, we remove it and get the overlap among three crawls of each profiles.

Finally, we apply the same method (§ III-B) to detect CNAME cloaking-based tracking among these profiles.

Figure 10 shows the detection percentage of the CNAME cloaking-based tracking among browsers and their extensions. Overall, all browsers and extensions have a different impact on CNAME cloaking-based tracking. The most aggressive browser is Brave. It has the best performance among five browsers without any extension and blocks around 50% of websites that use CNAME cloaking-based tracking. We speculate that Shields feature (§ II-A2) is effective at detecting CNAME cloaking-based tracking. We also manually confirm that Shields blocks some CNAME cloaking-related subdomains, such as smetrics.10daily.com.au (Adobe), f7ds.liberation.fr (Eulerian), and 5ijo.01net.com (Eulerian).

For all browsers, the most effective extension is uBlock Origin that reduces around 70% of the websites containing CNAME cloaking. Adblock and Adblock Plus provide low protection abilities for all browsers. This result is not surprising because these extensions target ad-blocking. Another notable point is that uBlock Origin version 1.24.5rc1 with anti-CNAME cloaking-based tracking technique is better than uBlock Origin version 1.24.4. It however only impacts to Firefox browser because other browsers do not provide an API that allows an extension to perform DNS lookups [57].

VI. A Machine Learning Approach for Detecting CNAME Cloaking-based Tracking

Next, we describe our supervised machine learning-based approach to detect CNAME cloaking-based tracking.

A. Method overview

Figure 11 shows an overview of our method consisting of four steps: data preparation, feature extraction, model development, and evaluation.

[2]Atrica currently supports chromium-based and Firefox-based browsers.

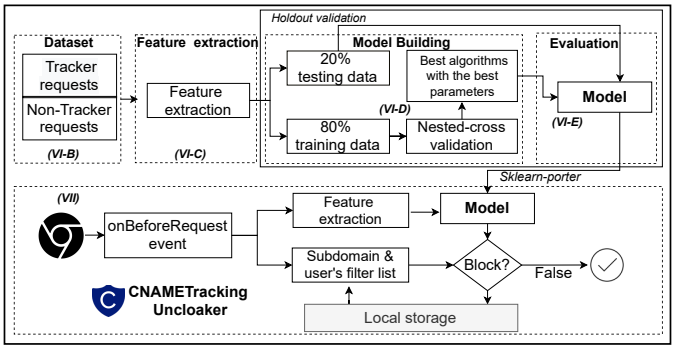

Fig. 11. Overview of machine learning approach for detecting CNAME cloaking-based tracking requests and CNAMETracking Uncloaker browser extension management workflow.

1) Select and divide dataset into two sets, which we call the tracker requests and the non-tracker requests (§ VI-B).
2) Extract features for all requests by subdomain (§ VI-C).
3) Compare the F1 score of 10 classification algorithms using 10-fold stratified nested cross-validation with over-sampled training data . After evaluating performance, we select the most effective classification algorithms with its best parameters to build a model (§ VI-D).
4) Evaluate the model with the testing data (§ VI-E).

B. Data preparation

We rely on 1,739 sites from Alexa Top 300K where CNAME cloaking-based tracking was previously detected § IV-C and another 1,739 sites randomly picked from these 300K sites without CNAME cloaking-based tracking from. We label all requests as tracker instances and non-tracker instances.

To analyze the concept drift of our model (see § VI-E), we also pick up 43,426 subdomain-related requests which belong to 1,009 sites are related to CNAME cloaking-based tracking and 1,009 additional randomly picked sites without CNAME-cloaking from April 2018.

The details of the 2020 dataset and the 2018 dataset are listed in Table VII.

TABLE VII
SUMMARY OF DATA: 2,010 SITES IN 2018 AND 3,524 SITES IN 2020.

Class	April 2018	January 2020
Tracker requests	2,490 (5.73%)	3,484 (10.01%)
Non-Tracker requests	40,939 (94.27%)	31,328 (89.99%)
Total subdomain-related requests	43,429 (100%)	34,812 (100%)
Total sites	2,018 (100%)	3,478 (100%)

C. Feature extraction

We experimentally extract the following features related to request linked to CNAME cloaking-related tracking.

- *method*: The desired action to be performed for a given request. We hypothesize that the GET method is usually used for subdomain-related requests linked to CNAME cloaking.
- *is_xhr*: The request uses an API that provides scripted client functionality for transferring data between a client and a server. We hypothesize that CNAME cloaking requires making HTTP requests in JavaScript between client and tracking provider server.
- *content_type*: The HTML tag that resulted in a request, such as image, javascript, or document, which are defined in this IDL [58]. We hypothesize that a specific resource is fetched in a web request for CNAME cloaking purpose (script).
- *len_url*, *len_sub*, and *len_prefix_sub*: The length of request URL, subdomain, and subdomain prefix. We hypothesize that there is a dissimilarity between the length of functional resources and CNAME cloaking resources.
- *num_prefix_sub*: The number of subdomain prefixes. We hypothesize that website's publishers use only one prefix to create a subdomain to deploy CNAME cloaking-based tracking.
- *prefix_sub_blacklist*: The subdomain prefix is among subdomain prefixes in tracking filter lists [19] [35]. We hypothesize that website's publishers use the same keyword (that already in the blocklist) to create a subdomain to deploy CNAME cloaking-based tracking.
- *is_sub_dic*: The prefix of subdomain is a word in the English dictionary. We hypothesize that web publishers use random string as a subdomain to redirect to the tracking provider via CNAME record instead of meaningful keywords.
- *entropy_url*, *entropy_sub*, and *entropy_prefix_sub*: The randomness of request URL by calculating the metric entropy from request URL, subdomain, and subdomain prefix. We hypothesize that there are differences in the metric entropy between functional request URL, subdomain, and subdomain prefix and CNAME cloaking resources.

D. Modeling and preliminary results

Using holdout validation method, we first split the 2020 dataset (Table VII) into testing data and training data. The percentage of the data held over for testing is 20%. It is used in § VI-E to evaluate our model. Next, we describe how to build a classification model to detect CNAME cloaking-based tracking using testing data (80% of the 2020 dataset).

1) Model nested cross-validation: To perform hyperparameter optimization and model selection, while overcoming the problem of training dataset overfitting and the unbalanced nature of our dataset, we perform nested cross-validation using ADASYN algorithm [59]. We first use an outer 10-folds cross-validation loop to randomly split the training dataset into 10 smaller sets (folds) without replacement, where nine folds are used for the model training and the remaining one fold is for validating. We also use an inner loop to optimize the hyper-parameters of each model for each training dataset made of nine outer-folds. Note that, to evaluate the cross-validation with real data, we only conduct over-sampling on the minority class by applying ADASYN algorithm in the training folds and not in the validation folds. We perform a

TABLE VIII
BEST PARAMETERS OF SELECTED ALGORITHM (EXTRA TREES) FROM THE TRAINING PHASE REGARDING F1 SCORE.

Algorithm	Parameter	Value
Extra Trees	max_features	10
	min_samples_split	2
	min_samples_leaf	1
	bootstrap	False
	n_estimators	100

grid search optimization for this classification regarding the F1 score. After obtaining 10 performance estimates by repeating this procedure ten times, we take their average as the final performance estimate.

To deploy this machine learning model as browser extension easily and effectively (see § VII), we first decide to compare 10 popular classification algorithms and evaluate their F1 score using above stratified nested cross-validation procedure on the training data.

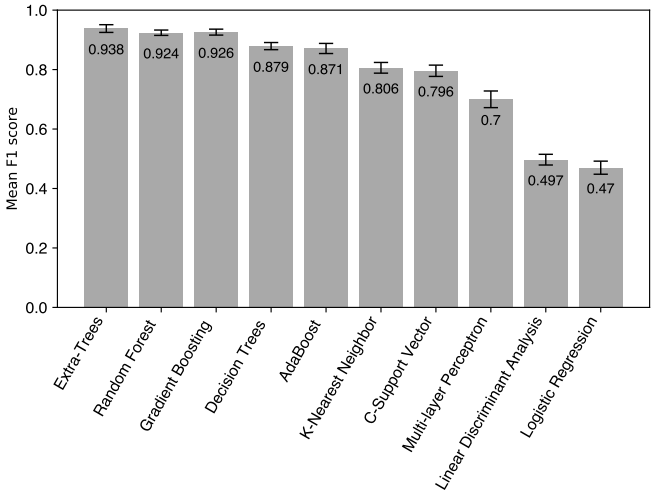

Fig. 12. F1 score for the 10 selected classification algorithms using 10-fold stratified nested cross-validation in the dataset regarding the detection of request linked to CNAME cloaking-based tracking. The mean and standard deviation are computed on the 10 folds of the nested cross-validation.

We use the F1 score for evaluating the performance of the classifiers. Larger values of the F1 score (≈ 1.0) indicate better performance, and lower values (≈ 0) correspond to worse performance. Figure 12 shows the F1 scores for the 10 selected algorithms using 10-fold stratified cross-validation in the 2020 dataset for detecting requests linked to CNAME cloaking-related tracking. All classification algorithms have a different detection performance. The most effective classification algorithm is Extra Trees, while Logistic Regression and Linear Discriminant Analysis classifiers show the worst performance for this dataset.

2) Selection of best algorithms and best parameters: From the previous performance evaluation, we select *Extra Trees* classifier and its set of best parameters (shown in Table VIII) to train our model with oversampled training data.

E. Classification performance evaluation

TABLE IX
A COMPARISON OF DETECTION PERFORMANCE.

Class	Method	Precision	Recall	F1 score
Non-Tracker	EasyPrivacy list	0.975	0.987	0.981
	Adguard filter list	0.938	0.996	0.966
	Filter lists + DNS API (uBO)	1.000	0.984	0.992
	Our approach	**0.991**	**0.997**	**0.994**
Tracker	EasyPrivacy list	0.866	0.777	0.819
	Adguard filter list	0.926	0.410	0.568
	Filter lists + DNS API (uBO)	0.877	1.000	0.934
	Our approach	**0.970**	**0.914**	**0.941**

1) The performance of model: The results obtained using the test set for requests linked to CNAME cloaking-based tracking detection on 20% of the 2020 dataset (preserved for this test) are shown in Table IX. We first show that our method detects requests related to CNAME cloaking-based tracking effectively. We achieve 0.941 of F1 score for tracker requests, 0.994 for non-tracker requests. We also obtain high precision and recall for both classes, which reduce the functional resources blocked by false positive, but still detect CNAME cloaking resources with less false negative.

By manually analyzing some false negatives and some false positives, we find that requests linked to CNAME cloaking have the same attributes as requests without CNAME cloaking based tracking. For example, a script *https://ea.hofmann.es/eahof4645.js* that points to tracking provider *Eulerian*; its prefix *ea* not in the blacklists and it looks like a functional script. However, we can classify initiated the request by this script that actually perform tracking behavior as CNAME cloaking-based tracking. On the contrary, the request by subdomain *pf.newegg.com* is not used for CNAME cloaking. Its request URL contains detailed tracking of user actions (including browser, device, and IP location), which make the length of this URL request is longer than these other functional requests. However, this request is also blocked by the Easy Privacy lists.

2) Comparison with other approaches: To block CNAME cloaking-based tracking without DNS resolution, well-known tracking filter lists such as the EasyPrivacy list and the AdGuard tracking filter list include the first-party subdomains which are fronts for CNAME cloaking. We thus compare the request detection performance between our machine learning approach and these well-known tracking filter lists in Table IX. We confirm lower F1 scores of tracker instances due to low recalls because of the lack of CNAME information for the tracking filter lists.

Furthermore, we simulate the performance of uBlock Origin with DNS API by applying CNAME resolution to the requests then matching them with three tracking filter lists (EasyPrivacy, Peter Lowe's tracking and uBlock Origin's own filter lists) as uBlock Origin does (Filter lists + DNS API in the table). As expected, filter lists with DNS API achieves the best performance. The reason of this high performance is due to the absence of false negatives (i.e., recall is 1.0) which is not the case for other methods, though the number of false positives is small (i.e., precision is ≈ 0.9). The performance of our ML approach without DNS API is here close to the best performance thanks to the trained model.

3) Feature permutation importance: To discover discriminative features for the detection, we investigate the permutation importance [60] to calculate the feature importance of selected classifier for a our dataset. Note that, larger values indicate higher importance.

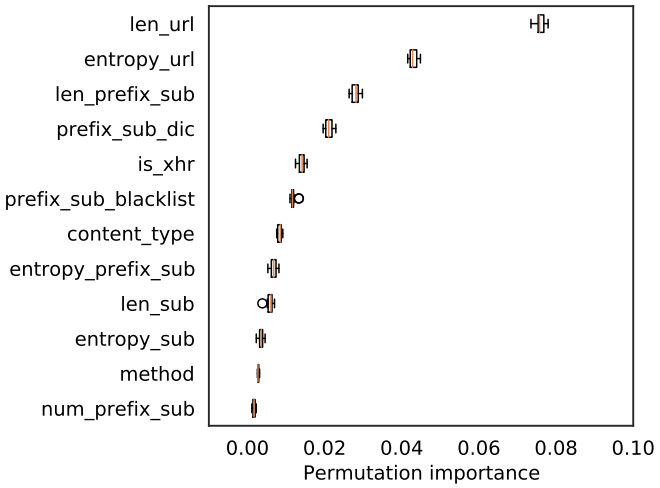

Fig. 13. Permutation importance of the selected model for CNAME cloaking-related tracking occurring. The box extends from the lower to upper quartile values of the data, with a line at the median. The number of times a feature is randomly shuffled is n_repeats = 10.

Figure 13 shows the feature permutation importance of the model for detecting the requests. The result reveals that the request URL length (*len_url*) has the highest importance. We assume that almost all requests with a subdomain used for CNAME cloaking-based tracking have a length longer than requests used for collecting content of site, because they contain users' identification. Also, the randomness of URL request (*entropy_url*) and the subdomain prefix length (*len_prefix_sub*) are discriminative features for request detection. In addition to that, the subdomain prefix blacklist presence (*prefix_sub_blacklist*) is not effective to detect request-related to CNAME cloaking. This is due to the fact that some publishers also use the same subdomain prefix that is used for both CNAME cloaking and other non-tracking resources.

4) Concept drift analysis: Finally, we investigate the ability to detect requests related to CNAME cloaking-based tracking in the latest dataset (2020) using a model trained on the old dataset (2018). We use the 2018 dataset to train a model and test it on new sites collected in 2020 (Table VII). We apply the method explained in § VI-D to build a model. Our result shows that the F1 score for CNAME cloaking-related requests detection is 0.703. Specifically, after two years, the F1 score decreases by 0.238. To explain this degradation, we examine the 2018 and 2020 datasets. In 2018, we do not see many random subdomain prefixes and short requests, while it is the case in 2020. These changes can be plausible reasons for the degradation of our model. Besides that, with rapid changes of web technology, tracking providers might also adjust their target site and change the implementation methods to deploy CNAME cloaking-based tracking Although the performance degradation is limited between 2018 and 2020, periodic model retraining can alleviate this problem if more detection accuracy is required.

VII. CNAMETracking Uncloaker - A machine learning-based browser extension to protect end-user from CNAME cloaking-based tracking

Through the comprehensive analysis above, we demonstrate the effectiveness of our model to classify request linked to CNAME cloaking-based tracking.

In this section, we propose *CNAMETracking Uncloaker*, a prototype of a Google Chrome browser extension that combines the blocklisting technique with the supervised machine learning for the automatic classification and filtering of CNAME cloaking-based tracking. As far as we know, only Firefox allows uBlock Origin to block CNAME cloaking by performing a DNS lookup of the hostname loading a resource. Other browsers do not support such DNS resolution API [8] (see § VIII-B). Our prototype implementation thus circumvents the lack of DNS API in Chrome-based browsers.

A. Design and implementation

The objective of our method is to monitor the HTTP requests and block a request if it is related to CNAME cloaking-based tracking. To this end, we intercept all subdomain-related request in-flight. In our extension, we track the HTTP requests and apply a machine learning model and a specific subdomain blocklist to detect and block CNAME cloaking-related requests. The novelty of our approach is combine the well-known technique of filtering with machine learning techniques to automatize the overall process.

Firstly, we use sklearn-porter [61] to transpile trained ExtraTree estimators (see § VI-D2) to JavaScript. To remove all unnecessary characters from JavaScript source code without altering its functionality, we also minify this file using UglifyJS [62]. We also build a feature extraction module on extension, which contains all 12 features described in § VI-C.

Secondly, we build a CNAME cloaking-based tracking subdomain filter list based-on our dataset on § IV-A. When this blocklist is updated on the server-side, *CNAMETracking Uncloaker* obtains an updated blocklist for CNAME cloaking-related subdomains.

Next, we implement the interface for accessing the user interface and changing the custom blocklist and allowlist. This interface allows end-users to construct customized lists according to the user's browsing habits; each user can have his/her own different configuration. The end-users can also define an allowlist to add exceptions to reduce the effect of machine learning-related false positives.

In the end, to intercept all HTTP requests by subdomain, we use the *onBeforeRequest* event, which is sent before any TCP connection is made and can be used to cancel or redirect requests. We then apply the feature extraction module and predict CNAME cloaking using the model. In addition, we match any subdomain-related URL request by CNAME cloaking-based tracking subdomain filter list and the custom filter lists of the end-user. If any subdomain-related request is not in the allowlist but it predicted as CNAME cloaking by model or by these blocklists, *CNAMETracking Uncloaker* blocks this request and keeps this element from being loaded onto the page by using a blocking event handler.

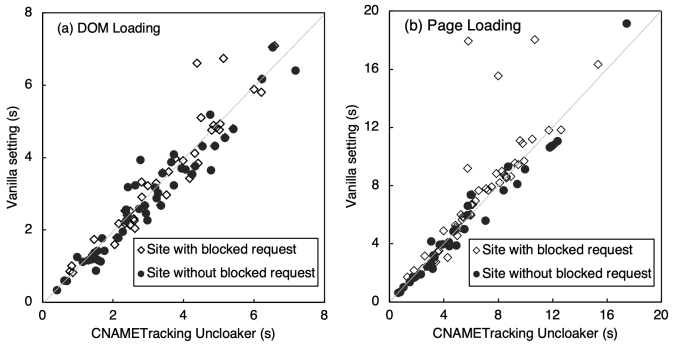

Fig. 14. Comparison of the performance overhead by CNAMETracking Uncloaker and vanilla setting based on Alexa Top 50 websites and 50 websites containing CNAME cloaking-based tracking on median delay to the DOMContentLoaded event and page load overhead in 10 times.

B. Performance evaluation

In order to measure the performance overhead of our extension, we compare it to the vanilla setting Chrome browser. We visit the top 50 websites according to Alexa's ranking (which do not contain CNAME cloaking-based tracking) and 50 websites linked to CNAME cloaking-based tracking in two cases: with and without *CNAMETracking Uncloaker*. During each visit, we record times of the *DOMContentLoaded* and *Load page* events. Each test is repeated 10 times with clean browsing history to retrieve the median duration. We use a commodity laptop computer with 8GB of RAM, Intel's i7 CPU, Ubuntu 18.04 and the latest version of Google Chrome (version 84.0). We present the performance overhead as the time difference between the median page load time with and without our extension.

Figure 14 (a) and (b) compare the overhead for *CNAMETracking Uncloaker* and the vanilla setting browser of the load events. We then plot two combinations: websites with blocked requests (blue circles) and without blocked requests (orange rectangles) by our extension. Our experiment shows that the overhead of *CNAMETracking Uncloaker* is acceptable. For the median gain for these sites, *CNAMETracking Uncloaker* extension introduces 0.08 seconds of median delay to the DOM loading. However, it is 0.408 seconds faster to the overall page loading, especially for websites that contain CNAME cloaking-based tracking. In particular, there are a limited number of websites without blocked requests (black circle) takes longer than the vanilla setting. Meanwhile, our extension blocks some resources (white rectangles) that reduce the time load site, this is why the overhead of *CNAMETracking Uncloaker* is lower than the other.

Interestingly, we observe that the vanilla setting is slower for three particular websites than *CNAMETracking Uncloaker*: *sohu.com* (12.15 seconds), *sina.com.cn* (7.54 seconds), and *clickavia.ru* (7.36 seconds). The reason for this behaviour can be explained by the large number of HTTP requests that were blocked by our extension for these three sites: *sohu.com*, *sina.com.cn*, and *clickavia.ru* (seven, seven, and one request(s), respectively). Note that for *sohu.com*, these requests are not CNAME cloaking based tracking. However, when we inspect these individual URLs using EasyPrivacy list as a reference, we can label these URLs as tracking-related. For *sina.com.cn*, one request is linked to tracking, but six requests are unrelated to CNAME cloaking. We also manually confirm that this website still works correctly, and that users can use our interface to customize the filter list (allowlist) to add an exception to our model.

In summary, *CNAMETracking Uncloaker* is able to filter out CNAME cloaking-based tracking from users' requests without significant performance degradation when compared with the default setting on Chrome browser. Note that *CNAMETracking Uncloaker* was only tested in our laboratory, we intend to do Beta testing of a group of target users to evaluate product performance in the real world in future work.

VIII. Related Work

A. Third-party tracking detection and privacy protection technique comparison

The privacy hazards of online web tracking have been studied extensively.

1) Machine learning-based tracking detection: Many machine learning-based approaches have been proposed to detect third-party web tracking automatically. Metwalley et al. [13] developed an unsupervised detection method that inspects URL queries in HTTP(S) requests to detect tracking activities. Yamada et al. [63] analyzed traffic at the network gateway to monitor all tracking sites in the administrative network and constructs a graph between sites and their visited time to detect tracking sites. Wu et al. [12] developed DMTrackerDetector which automatically detects third-party trackers offline to efficiently generate blocklists using structural hole theory and supervised machine learning. Ikram et al. [64] proposed one-class machine learning classifiers using syntactic and semantic features extracted from JavaScript programs to classify functional and tracking JavaScript programs.

2) Non-machine learning-based tracking detection: In addition, some studies focus on a particular way to detect third-party web tracking. Schelter and Kunegis [65] performed a large-scale analysis of third-party trackers by extracting third-party embeddings from more than 41 million domains to study global online tracking. Roesner et al. [66] developed a client-side method for detecting and classifying five types of third-party trackers over 500 unique trackers on the 500 most popular and 500 less popular sites according to the Alexa ranking sites. To cut off the tracking chain of third-party web tracking, Pan et al. [67] developed TrackingFree which isolates unique identifiers into different browser principals so that the identifiers still exist but are not unique among different websites.

3) Privacy protection technique comparison: Besides, many researches measure privacy protection techniques. In previous studies, Mazel et al. [29] proposed a reliable methodology for privacy protection techniques comparison and compared a wide range of privacy protection techniques. Ruiz-Martinez [68] presented a survey of the theoretical comparison of the solutions and the main tools for privacy concern when users and surfing on the Internet. Mayer et al. [4] surveyed the current policy debate surrounding third-party web tracking and explains the relevant technology and uses a fourth-party web

measurement platform to collect HTTP requests, responses and cookies. Merzdovnik et al. [20] performed a measurement study on the effectiveness of popular tracker-blocking tools on more than 100,000 popular websites and 10,000 popular Android applications.

Our work not only focuses on privacy extensions but also extensively compares a wide set of browsers and privacy protection techniques against a certain third-party tracking - CNAME cloaking-based tracking.

B. CNAME cloaking-based tracking

1) CNAME cloaking measurement: In the uBlock Origin's GitHub issues page, a user presented a website loading first-party request, that pointed to a tracking provider [57]. This issue was then addressed in several discussions [15] [69] [70].

After our original work [1], several studies reported more aspects of CNAME cloaking [71]–[73]. Aliyeva and Egele [71] and Ren et al. [72] focused on the security aspect by analyzing the impact of CNAME cloaking on browser cookie policies, which may transmit sensitive cookies to third parties. Furthermore, Dimova et al. [73] reported on a large-scale longitudinal evaluation CNAME cloaking-based tracking using the HAR dataset, by detecting five tracking providers by using a three-pronged approach and extended the list with eight trackers from the CNAME cloaking blocklist by NextDNS [74]. Besides the security analysis, they presented consistent results with ours, especially sites containing CNAME cloaking is gradually increasing over time. Here, our study concentrated on privacy view, not only characterized this phenomenon, but also provided a wider picture of current privacy protection techniques, by evaluating the effect of well-known filters, browsers, and extensions against CNAME cloaking-based tracking.

2) CNAME cloaking counter-measures: Several counter-measures have been developed to protect end-user from CNAME cloaking-based tracking, including network-based DNS blocking and in-browser techniques.

Network-based blocking methods are in use before web browsers support the conception of extensions [20]. NextDNS [15] and AdGuard [16] are applications at the DNS level, which require the wildcard match (domain and all its multi-level subdomains) against the domains in the CNAME cloaking blocklist. Nevertheless, NextDNS is a commercial product and it requires to install and configure the NextDNS client; AdGuard DNS is free in personal use, but end-users must set up their DNS servers and send their entire DNS traffic to the AdGuard server. In addition, Pi-hole [17] is a DNS sinkhole that protects end-users' devices from unwanted contents. However, the end-users have to install a supported operating system and Pi-hole on user's devices or separate hardware/appliance, then configure users router's DHCP options to force clients to use Pi-hole as their DNS server.

The in-browser privacy protection techniques not only improve users' privacy but can also increase users' browsing experience [75]. To make sure these potential advantages, some browser extensions also update themselves to block CNAME cloaking-based tracking resources. Adguard blocker [76], uBlocker Oringin [8], make an continuous effort to manually update first-party subdomains which are fronts for CNAME cloaking to these blocklists. It makes day-to-day filter lists updating tedious and time-consuming. As we evaluated in § V, these extensions show the moderate detection performance to detect CNAME cloaking, except uBlock Origin with DNS API only supported by the Firefox browser. To keep up with this tracking technique, the Safari and Brave add a new feature that keeps their users protected. The ITP Safari lowers the duration of cookies set in the HTTP response created through JavaScript to defense with CNAME cloaking [77]. Meanwhile, the Brave embedded DNS resolver to block any request that has the canonical domain in their blacklist by default [78]. However, these browsers account for a small percentage of browser share [52].

To overcome these constraints, we propose an in-browser counter-measure that is based on a supervised machine learning-based method and a subdomain blocklist to detect CNAME cloaking-based tracking without the on-demand DNS lookup. To the best of our knowledge, we here propose the first in-browser extension relies on machine learning techniques to protect the end-user from CNAME cloaking-based tracking (see Table X).

TABLE X
DETAILED COMPARISON OF OUR COUNTER-MEASURE WITH OTHER RELEVANT WORKS AVAILABLE IN LITERATURE.

Counter-measure	subdomain blocking list	DNS & blocking list	Machine learning	Cookie configuration	in-browser technique
NextDNS [15]		✓			
AdGuard DNS [16]		✓			
Pi-hole [17]		✓			
AdGuard blocker [76]	✓				✓
uBlock Origin (Firefox) [79]	✓	✓			✓
uBlock Origin [8]	✓				✓
ITP (Safari) [80]				✓	✓
Shields (Brave) [23]		✓			✓
Our work	✓		✓		✓

IX. LIMITATIONS

Although we have expanded a great deal of effort in our study, there are still some drawbacks.

Firstly, our *CNAME tracking filter list* in blocklist-based detection approach may be incomplete. We rely on the Easy Privacy list and Adguard Tracking filter list that are well-known and widely-used over the years, both by end-users and as ground-truth in academic works [81]–[83]. However, if the tracking providers use new domains that do not belong to these filter lists, we might miss these cases. Comparing with the lists of trackers that are often disguised using CNAME that publish by Adguard [84] and NextDNS [74], we observe that we dismissed four tracking providers: MO Internet Group, GENIEE, TraceDock, and Lead Forensics. However, these tracking providers did not appear in our dataset. Secondly, we observed some unstable crawling results even in our three crawls per site for in-browser protection techniques comparison. We omitted unfinished crawling results due to timeout, but still there is a possibility to miss CNAME cloaking resources because of the package loss or the rapid change of web content. Finally, our dataset was crawled from a single country (Japan or USA), so there is a possibility of causing geographical differences.

X. Conclusion

In this paper, we characterized, detected, and protected the end-user against CNAME cloaking-based tracking on the web.

We conducted experiments to assess the occurrence and evolution of CNAME cloaking-based tracking. The results show that 1,739 websites in the Alexa Top 300K sites in January 2020 contain CNAME cloaking-based tracking. These websites are spread across many countries and categories. We also characterized a longitudinal analysis of CNAME cloaking-based tracking from 2016 to 2020. We found a significant evidence that the top websites have injected more CNAME cloaking-based tracking in the last four years. The current best counter-measure to defend CNAME cloaking-based tracking, blocklist approach, is strongly depend on realtime name resolution. To overcome this limitation, we proposed a machine learning approach to detect HTTP requests containing CNAME cloaking-based tracking. Through the comprehensive analysis, we demonstrate the effectiveness of our method. Meanwhile, the DNS API being only available in Firefox browser, we developed a browser Chrome extension *CNAMETracking Uncloaker*, exploiting this machine learning-based approach to classify CNAME cloaking-based tracking. We performed an exhaustive evaluation of performance and effectiveness of our software prototype showing that machine learning-based techniques can be employed client-side as solutions into the browser to protect end-users against CNAME cloaking-based tracking.

Overall, the contributions of this paper advance the field of web privacy by providing not only the largest study on the CNAME cloaking-based tracking but furthermore, proposing the machine learning-based approach for detecting CNAME cloaking, along with the prototype implementation in a browser extension to protect end-users from this kind of web tracking.

Dataset availability: We provide a list of CNAMEs and tracking providers using CNAME cloaking based tracking in our analysis at https://github.com/fukuda-lab/cname_cloaking. The extension is publicly available at Chrome Store: [85]. The raw crawled dataset will be also available from the authors on request.

References

[1] H. Dao, J. Mazel, and K. Fukuda, "Characterizing cname cloaking-based trackingon the web," in *Proceedings of IFIP/IEEE TMA*, 2020, pp. 1–9.

[2] H. Dao and K. Fukuda, "A machine learning approach for detecting cname cloaking-based tracking on the web," in *Proceedings of IEEE GLOBECOM*, 2020, pp. 1–6.

[3] (2016) 2016 truste/ncsa consumer privacy infographic - gb edition. [Online]. Available: https://www.trustarc.com/resources/privacy-research/ncsa-consumer-privacy-index-gb/

[4] J. R. Mayer and J. C. Mitchell, "Third-party web tracking: Policy and technology," in *IEEE S&P*, 2012, pp. 413–427.

[5] Adblock. [Online]. Available: https://getadblock.com/

[6] Adblock plus. [Online]. Available: https://adblockplus.org/

[7] Disconnect. [Online]. Available: https://disconnect.me/

[8] R. Hill. ublock origin - an efficient blocker for chromium and firefox. fast and lean. [Online]. Available: https://github.com/gorhill/uBlock

[9] V. Dudykevych and V. Nechypor, "Detecting third-party user trackers with cookie files," in *International Scientific-Practical Conference Problems of Infocommunications Science and Technology (PIC S&T)*. IEEE, 2016, pp. 78–80.

[10] G. Acar, C. Eubank, S. Englehardt, M. Juarez, A. Narayanan, and C. Diaz, "The web never forgets: Persistent tracking mechanisms in the wild," in *Proceedings of ACM SAC*, 2014, pp. 674–689.

[11] S. Englehardt and A. Narayanan, "Online tracking: A 1-million-site measurement and analysis," in *Proceedings of ACM CCS*, 2016, pp. 1388–1401.

[12] Q. Wu, Q. Liu, Y. Zhang, P. Liu, and G. Wen, "A machine learning approach for detecting third-party trackers on the web," in *Proceedings of ESORICS*, 2016, pp. 238–258.

[13] H. Metwalley, S. Traverso, and M. Mellia, "Unsupervised detection of web trackers," in *Proceedings of IEEE GLOBECOM*, 2015, pp. 1–6.

[14] N. Kushmerick, "Learning to remove internet advertisements," in *Proceedings of AGENTS'99*, 1999, pp. 175–181.

[15] O. Poitrey. (2019) Nextdns first to support blocking of all third-party trackers disguised as first-party. [Online]. Available: https://medium.com/nextdns/nextdns-added-cname-uncloaking-support-becomes-the-first-cross-platform-solution-to-the-problem-e3f437f84342

[16] Adguard dns. [Online]. Available: https://adguard.com/en/adguard-dns/overview.html

[17] Pi-hole – a black hole for internet advertisements. [Online]. Available: https://pi-hole.net/

[18] Privacy badger — electronic frontier foundation. [Online]. Available: https://www.eff.org/privacybadger

[19] Adguard tracking protection filter. [Online]. Available: https://filters.adtidy.org/extension/chromium/filters/3.txt

[20] G. Merzdovnik, M. Huber, D. Buhov, N. Nikiforakis, S. Neuner, M. Schmiedecker, and E. Weippl, "Block me if you can: A large-scale study of tracker-blocking tools," in *IEEE EuroS&P 2017*, 2017, pp. 319–333.

[21] Ghostery makes the web cleaner, faster and safer! [Online]. Available: https://www.ghostery.com/

[22] Firefox browser. [Online]. Available: https://www.mozilla.org/en-US/exp/firefox/

[23] Brave browser. [Online]. Available: https://brave.com/

[24] Tor browser. [Online]. Available: https://www.torproject.org/

[25] (2019) Today's firefox blocks third-party tracking cookies and cryptomining by default. [Online]. Available: https://blog.mozilla.org/blog/2019/09/03/todays-firefox-blocks-third-party-tracking-cookies-and-cryptomining-by-default/

[26] (2008) The design and implementation of the tor browser [draft]. [Online]. Available: https://2019.www.torproject.org/projects/torbrowser/design/

[27] (2019) Data collection cnames and cross-domain tracking. [Online]. Available: https://docs.adobe.com/content/help/en/id-service/using/reference/analytics-reference/cname.html

[28] The top 500 sites on the web. [Online]. Available: https://www.alexa.com/topsites

[29] J. Mazel, R. Garnier, and K. Fukuda, "A comparison of web privacy protection techniques," *Computer Communications*, vol. 144, pp. 162–174, 2019.

[30] Q. Scheitle, O. Hohlfeld, J. Gamba, J. Jelten, T. Zimmermann, S. D. Strowes, and N. Vallina-Rodriguez, "A long way to the top: significance, structure, and stability of internet top lists," in *Proceedings of ACM IMC*, 2018, pp. 478–493.

[31] Http archive. [Online]. Available: https://httparchive.org/

[32] Public suffix list. [Online]. Available: https://publicsuffix.org/list/

[33] Rapid7 open data, forward dns (fdns). [Online]. Available: https://opendata.rapid7.com/sonar.fdns_v2/

[34] Dnsdb database. [Online]. Available: https://www.dnsdb.info/

[35] Easyprivacy. [Online]. Available: https://easylist.to/easylist/easyprivacy.txt

[36] Marketing attribution and data management - eulerian. [Online]. Available: http://www.eulerian.com/

[37] S. Kayan. cdnfinder. [Online]. Available: https://github.com/turbobytes/cdnfinder/blob/master/assets/cnamechain.json

[38] W. Ma. china-cdn-domain-whitelist. [Online]. Available: https://github.com/mawenjian/china-cdn-domain-whitelist/blob/master/china-cdn-domain-whitelist.txt/

[39] Z. Weinberg, S. Cho, N. Christin, V. Sekar, and P. Gill, "How to catch when proxies lie: Verifying the physical locations of network proxies with active geolocation," in *Proceedings of ACM IMC*, 2018, pp. 203–217.

[40] Ip geolocation api. [Online]. Available: https://ip-api.com/

[41] Free ip geolocation api. [Online]. Available: https://freegeoip.app/

[42] Maxmind geoip2 python api. [Online]. Available: https://dev.maxmind.com/geoip/geoip2/geolite2/

[43] Fortiguard web filtering. [Online]. Available: https://fortiguard.com/webfilter

[44] Tracking protection lists - trackers we block. [Online]. Available: https://github.com/mozilla-services/shavar-prod-lists/blob/master/disconnect-blacklist.json

[45] Pardot —b2b marketing automation. [Online]. Available: https://www.pardot.com/

[46] Act-on — exceptional marketing automation software. [Online]. Available: http://www.act-on.com/

[47] Oracle eloqua marketing automation. [Online]. Available: https://www.oracle.com/cx/marketing/automation/

[48] Webtrekk. [Online]. Available: https://www.webtrekk.com/

[49] General data protection regulation - gdpr. [Online]. Available: https://gdpr-info.eu/

[50] Python parser for adblock plus filters. [Online]. Available: https://github.com/scrapinghub/adblockparser

[51] Adblockparser limitations. [Online]. Available: https://github.com/scrapinghub/adblockparser#limitations

[52] Browser statistics. [Online]. Available: https://www.w3schools.com/browsers/

[53] (2008) Chrome browser. [Online]. Available: https://www.google.com/chrome/

[54] Opera browser. [Online]. Available: https://www.opera.com/

[55] R. Hill. ublock origin - developer build 1.24.5rc0. [Online]. Available: https://github.com/gorhill/uBlock/releases/tag/1.24.5rc0

[56] V. Gerest. Atrica. [Online]. Available: https://github.com/fukuda-lab/atrica

[57] (2019) Address 1st-party tracker blocking #780. [Online]. Available: https://github.com/uBlockOrigin/uBlock-issues/issues/780#issuecomment-566845764

[58] base - mozsearch. [Online]. Available: https://searchfox.org/mozilla-central/source/dom/base/

[59] H. He, Y. Bai, E. A. Garcia, and S. Li, "ADASYN: Adaptive synthetic sampling approach for imbalanced learning," in *Proceedings of IEEE International joint conference on neural networks*, 2008, pp. 1322–1328.

[60] L. Breiman, "Random forests," *Machine learning*, pp. 5–32, 2001.

[61] D. Morawiec, "sklearn-porter," transpile trained scikit-learn estimators to C, Java, JavaScript and others. [Online]. Available: https://github.com/nok/sklearn-porter

[62] Uglifyjs is a javascript parser, minifier, compressor and beautifier toolkit. [Online]. Available: https://github.com/mishoo/UglifyJS

[63] A. Yamada, M. Hara, and Y. Miyake, "Web tracking site detection based on temporal link analysis," in *Proceedings of IEEE AINA Workshop*, 2010, pp. 626–631.

[64] M. Ikram, H. J. Asghar, M. A. Kaafar, A. Mahanti, and B. Krishnamurthy, "Towards seamless tracking-free web: Improved detection of trackers via one-class learning," *Proceedings on PET*, vol. 2017, no. 1, pp. 79–99, 2017.

[65] S. Schelter and J. Kunegis, "Tracking the trackers: A large-scale analysis of embedded web trackers," in *Proceedigns of AAAI Conference on Web and Social Media*, 2016.

[66] F. Roesner, T. Kohno, and D. Wetherall, "Detecting and defending against third-party tracking on the web," in *Proceedings of USENIX NSDI*, 2012, pp. 12–12.

[67] X. Pan, Y. Cao, and Y. Chen, "I do not know what you visited last summer protecting users from third-party web tracking with tracking free browser," in *NDSS'15*, 2015, pp. 1–16.

[68] A. Ruiz-Martínez, "A survey on solutions and main free tools for privacy enhancing web communications," *Journal of network and computer applications*, vol. 35, no. 5, pp. 1473–1492, 2012.

[69] R. Cointepas. (2019) Cname cloaking, the dangerous disguise of third-party trackers. [Online]. Available: https://medium.com/nextdns/cname-cloaking-the-dangerous-disguise-of-third-party-trackers-195205dc522a

[70] J. Leyden. (2019) Web trackers using cname cloaking to bypass browsers' ad blockers. [Online]. Available: https://portswigger.net/daily-swig/web-trackers-using-cname-cloaking-to-bypass-browsers-ad-blockers

[71] A. Aliyeva and M. Egele, "Oversharing is not caring: How cname cloaking can expose your session cookies," in *Proceedings of ASIACCS*, 2021, pp. 1–12.

[72] T. Ren, A. Wittman, L. De Carli, and D. Davidson, "An analysis of first-party cookie exfiltration due to cname redirections," in *Proceedings of MADWeb*, 2021, pp. 1–11.

[73] Y. Dimova, G. Acar, L. Olejnik, W. Joosen, and T. Van Goethem, "The cname of the game: Large-scale analysis of dns-based tracking evasion," *Proceedings on PET*, vol. 2021, no. 3, pp. 393–411, 2021.

[74] Nextdns cname cloaking blocklist. [Online]. Available: https://github.com/nextdns/cname-cloaking-blocklist/blob/master/domains

[75] K. Borgolte and N. Feamster, "Understanding the performance costs and benefits of privacy-focused browser extensions," in *Proceedings of WWW*, 2020, pp. 2275–2286.

[76] Adguard adblocker. [Online]. Available: https://adguard.com/en/adguard-browser-extension/overview.html

[77] Cname cloaking and bounce tracking defense. [Online]. Available: https://webkit.org/blog/11338/cname-cloaking-and-bounce-tracking-defense/

[78] What's brave done for my privacy lately? episode #6: Fighting cname trickery. [Online]. Available: https://brave.com/privacy-updates-6/

[79] ublock origin - firefox. [Online]. Available: https://addons.mozilla.org/en-US/firefox/addon/ublock-origin/

[80] Safari - apple. [Online]. Available: https://www.apple.com/safari/

[81] O. Starov and N. Nikiforakis, "Privacymeter: Designing and developing a privacy-preserving browser extension," in *Proceedings of ESSoS*. Springer, 2018, pp. 77–95.

[82] S. Englehardt, J. Han, and A. Narayanan, "I never signed up for this! privacy implications of email tracking," *Proceedings on PET*, vol. 2018, no. 1, pp. 109–126, 2018.

[83] Q. Chen, P. Snyder, B. Livshits, and A. Kapravelos, "Improving web content blocking with event-loop-turn granularity javascript signatures," *arXiv preprint arXiv:2005.11910*, 2020.

[84] Adguard cname original trackers list. [Online]. Available: https://github.com/AdguardTeam/cname-trackers/blob/master/combined_original_trackers.txt

[85] Cnametracking uncloaker. [Online]. Available: https://chrome.google.com/webstore/detail/cnametracking-uncloaker/fhhdlfepbipknmeclodhcapbkmpdehkb

Ha Dao is a Ph.D student at the Graduate University for Advanced Studies (SOKENDAI), Japan. Her research interests are in online privacy and data protection.

Johan Mazel is a researcher at ANSSI (French National Cybersecurity Agency). He received his PhD from INSA Toulouse/LAAS-CNRS in 2011. He was a post-doctoral researcher at the National Institute of Informatics (NII) in Tokyo, Japan from 2012 to 2016. His research interests focus on network traffic monitoring and analysis for security.

Kensuke Fukuda received the Ph.D. degree in computer science from Keio University, Kanagawa Japan, in 1999.

He is an Associate Professor with the National Institute of Informatics (NII) and the Graduate University for Advanced Studies (SOKENDAI). His research interests span measurement and analysis of Internet traffic, network management and security.