Oversharing Is Not Caring: How CNAME Cloaking Can Expose Your Session Cookies

Assel Aliyeva
Boston University
USA
aliyevaa@bu.edu

Manuel Egele
Boston University
USA
megele@bu.edu

ABSTRACT

In modern web ecosystem, online businesses often leverage third-party web analytics services to gain insights into the behavior of their users. Due to the recent privacy enhancements in major browsers that restrict third-party cookie usage for tracking, these businesses were urged to disguise third-party analytics infrastructure as regular subdomains of their websites [3]. The integration technique referred to as CNAME cloaking allows the businesses to continue monitoring user activity on their websites. However, it also opens up the possibility for severe security infractions as the businesses often share their *session cookies* with the analytics providers, thus putting online user accounts in danger.

Previous work has raised privacy concerns with regards to subdomain tracking and extensively studied the drawbacks of widely used privacy-enhancing browser extensions. In this work, we demonstrate the impact of deploying CNAME cloaking along with lax cookie access control settings on web user security. To this end, we built a system that automatically detects the presence of the disguised third-party domains as well as the leakage of the first-party cookies. Using our system, we identified 2,139 web analytics domains that can be conveniently added to commonly deployed host-based blacklists. Concerningly, we also found that 27 out of 90 highly sensitive web services (e.g., banks) that we analyzed expose session cookies to the web analytics services.

CCS CONCEPTS

• **Security and privacy** → **Access control**; Web application security.

KEYWORDS

cookies; access control; CNAME cloaking

ACM Reference Format:
Assel Aliyeva and Manuel Egele. 2021. Oversharing Is Not Caring: How CNAME Cloaking Can Expose Your Session Cookies. In *Proceedings of the 2021 ACM Asia Conference on Computer and Communications Security (ASIA CCS '21), June 7–11, 2021, Hong Kong, Hong Kong*. ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3433210.3437524

ASIA CCS '21, June 7–11, 2021, Hong Kong, Hong Kong

ACM ISBN 978-1-4503-8287-8/21/06…$15.00
https://doi.org/10.1145/3433210.3437524

1 INTRODUCTION

The number of active Internet users has been rapidly increasing in the recent decade and surged from 1.97 billion people in 2010 [11] to 4.57 billion in July 2020 [10]. Such growth dynamics incentivizes businesses to move online in order to expand the outreach of their products. Nowadays, businesses offer a wide range of online services, such as shopping, financial transactions, or virtual meeting platforms through their websites. Many websites also allow users to create online accounts. These accounts separate authenticated users from other website visitors and allow registered users to manage their private information as well as to access additional functionalities. Because of the sensitivity of the information stored in user accounts, the security of these accounts is essential: a successful account takeover may lead to severe financial losses and the leakage of sensitive data.

A crucial linchpin in the security of online accounts is the confidentiality of session cookies. In this work, we use the term *session cookies* to refer to the *cookies used to authenticate a specific user to a web service*. Obviously, these cookies must exclusively be known to the user and the web service. Any third-party with access to the cookies can easily impersonate the legitimate user on the web service[1]. As the first line of defense, many businesses deploy the HTTPS protocol that guarantees the confidentiality and integrity of HTTP communications (including session cookies) with their users.

In addition to serving their users, online businesses frequently also seek to leverage data of their visitors to generate additional profit (e.g., via advertising) or use analytics (or tracking) insights to improve their websites. The commoditization of web services resulted in dedicated companies (which we refer to as *T/A (tracking or advertising) services or T/As*) that provide this functionality. Until recently, T/As were hosted on third-party domains and used third-party cookies to re-identify the same user across the different websites they provided T/A services to. However, recent browser developments, mostly driven by user privacy concerns, resulted in default settings that reject third-party cookies [35, 36, 38]. This proves a decisive blow to the business model of established T/A services, prompting them to adapt their strategy. In reaction to the exclusion of third-party cookies, prominent T/As now urge their customers (i.e., the first-party websites) to configure their DNS settings to include the T/A service's infrastructure as a subdomain of the first-party. This inclusion happens via *CNAME cloaking* [3] (see Section 2.2 for details), a technique that relies on CNAME DNS

[1]While the OWASP recommends to tie session cookies to the client's IP[21], this mechanism does not seem to be widely deployed. A potential reason is the negatively impacted usability as roaming between different networks (e.g., home, public WiFi, and work) would require to login again.

records, similar to how content delivery networks have worked for years.

This approach of embedding T/A services as a first-party opens up the possibility for severe security infractions. Specifically, if this mechanism is combined with lax settings of the access controls on first-party cookies (see Section 2.1 for details), these cookies will leak to T/A services, and hence breach the confidentiality requirement for session cookies.

Prior research that focused on the implications of CNAME cloaking mainly raised privacy concerns after examining the sources of personal data leakage [25–27] and the (in-)effectiveness of the existing privacy-enhancing browser extensions [29, 30]. Other related works are unrelated to CNAME cloaking and shed light on the scale of data syncing mechanisms within the web analytics and advertisement ecosystem [31, 41, 45] and investigate how mixed HTTP/HTTPS content affects the privacy and security of user accounts [39, 43]. To the best of our knowledge, no prior work has investigated the eradication of vital security protections caused by the combination of the inclusion of T/A services via CNAME cloaking and prolifically lax access control of cookies.

In this work, we first identify the negative security outcomes that follow from using T/A services via CNAMEs combined with lax access control on cookies. Subsequently, we provide a mechanism and our system (TAFinder) to automatically detect such (presumably unwanted) leakage. Finally, we evaluate TAFinder to assess the prevalence of these issues via a large-scale measurement on the 100,000 most popular websites in the Majestic Million [12] rankings.

Our results are concerning, as 1,195 out of 2,271 websites that rely on T/A services leak their cookies. Moreover, concrete case studies on banking and financial websites show how these entities mis-handle the security of their online accounts in the presence of cloaked T/A providers. We chose these business categories as they contain the most sensitive information and often allow users to manage their funds. While our experiments revealed some first-party services that avoid this problem by setting proper access control restrictions on their cookies (e.g., `discover.com`), other popular institutions (e.g., banks), fail to do so, and continue to do so even after we reported our findings to them. In a few extreme cases (437 websites), where T/A content is fetched over plain-text HTTP, this behavior even obviated the protections afforded by TLS/SSL, as on-path attackers get access to any first-party cookie that does not have its `Secure` attribute set. We manually verified the findings of TAFinder on the websites of 27 popular e-commerce and financial institutions (using our own accounts), confirmed the leakage of session cookies, and verified successful session hijacking for all of them.

To broaden TAFinder's knowledge of T/A services beyond commonly known actors (e.g., those found on blacklists), TAFinder includes a supervised machine learning approach which in turn uses nine features extracted from the HTTP communications a browser triggers when visiting a website. The results from this classification are independently valuable as the newly identified T/A services can be conveniently added to host-based blacklists which are frequently used by privacy-enhancing browser plugins. In summary, this paper makes the following contributions.

- We identify the necessary conditions that will result in the leaking of first-party cookies to T/A service providers.
- Based on that insight, we build a system (TAFinder) that automatically identifies the presence of cloaked domains in a website and detects whether the website's cookies leak to these domains.
- We also propose an HTTP-based web analytics domain detection mechanism for hidden domains that achieves the accuracy of 96 %.
- We identified 2,139 subdomains that disguise web analytics services that were not present in the hostname-based community blacklists.
- We evaluated TAFinder on the 100,000 most popular websites according to the Majestic Million dataset and found that 2,271 websites embed third-party analytics providers as their subdomains using the CNAME cloaking technique.
- Concerningly, even highly sensitive web services (e.g., banks) share their session cookies with the used T/A services, opening the door to trivial account takeover.
- We disclosed our findings to the affected vendors and received mixed responses including denial of the problem and silently fixing the reported issues.

2 BACKGROUND AND THREAT MODEL

In this section, we discuss commonly used web technologies relevant to our work. We also cover the basics of third-party tracking and advertising and details of the CNAME cloaking technique. Finally, we provide a threat model that captures the impact of CNAME cloaking and cookie access control settings on web user security and privacy.

2.1 Web essentials

HTTP cookies. Cookies are small chunks of data that websites store on a user's machine. This data typically contains various meta-information about the user, including her session identifiers, authentication tokens, location, etc. Websites pass cookies to the user in the HTTP response via the `Set-Cookie` header. Browsers store the received cookies in a cookie jar and attach them on every subsequent HTTP request to the website. In this way, the website can retain information about the user's session across multiple HTTP requests, even though the HTTP protocol is stateless. This makes cookies the prime choice to function as *authentication tokens*. Once the user authenticated herself against the web service, the service sets a secret session cookie that the browser will transparently include in subsequent requests to the website. Note that the security of the user's account crucially relies on the confidentiality of this session cookie (i.e., no third party beyond the user and the website should ever learn this value).

Cookies are characterized by their name, value, and a series of attributes. The cookie's name uniquely identifies it to the website while the value stores arbitrary information specified by the website. Each cookie also contains a series of attributes. The attributes most relevant to this paper are `Domain`, `HTTPOnly`, and `Secure`. The `Domain` attribute defines the so-called "scope" of the cookie. A cookie's scope is the set of hosts to which the browser will submit the cookie with any HTTP request. Essentially, the scope is an

access-control mechanism that dictates for each cookie whether it is included in an HTTP request. By default, the `Domain` property is empty, indicating that the cookie should only be included in requests targeting the precise host that set the cookie. However, website developers can choose to broaden the scope of the cookie by setting the `Domain` attribute to any ancestor-domain of the origin. In that case, the cookie will be included in requests to any domain *or subdomain* of the value specified in the `Domain` attribute. For example, by default a cookie received from `www.cnn.com`, will only be included in requests targeting `www.cnn.com`. However, to transmit the cookies to other cnn properties, the `Domain` attribute can be set to `cnn.com`. This will instruct browsers to include the cookie in all requests to any host under the `cnn.com` domain, including `www.cnn.com`, `mms.cnn.com`, and others.

Furthermore, the `HTTPOnly` attribute instructs the browser to exclusively include cookies in HTTP(S) requests while at the same time preventing scripts from accessing the cookie (e.g., through JavaScript's `document.cookie`). Finally, the `Secure` attribute instructs browsers to only include the such cookies in requests that are made through encrypted HTTPS connections, and omit the cookie otherwise.

Domain Name System. The most important responsibility of the Domain Name System (DNS) is to resolve human-readable domain names to numeric IP addresses. The DNS is organized as a distributed database, and its data entries, also known as *Resource Records*, are commonly defined by three attributes. A *type* attribute identifies the type of a given record. The *name* and *value* attributes are used in a key-value fashion. To associate a domain name $\mathcal{D}$ with an IP address, a DNS administrator simply adds the appropriate resource record of type `(A)` (Address Mapping) or `AAAA` (IP Version 6 Address), populates the *name* with $\mathcal{D}$, and the *value* attribute with the corresponding IP address. Of specific importance to this paper are resource records of type `CNAME` (Canonical Name). `CNAME` records introduce aliases into the DNS system and use domain names for both its *name* and *value* attributes. Essentially, a `CNAME` record creates an alias that instructs DNS resolvers (i.e., the clients of the DNS system) to resolve the domain listed as *name* as a different domain (i.e., *value*) instead. It is important to note that the domains provided in *name* and *value* need not be in the same administrative domain, or *Zone* in DNS terminology. Note that the same-origin policy and access controls via cookie's `Domain` attributes always rely on the domain that is used to refer to a resource. That is, if a resource is fetched from a host (i.e., the *value* attribute) which is an alias established via a DNS `CNAME` record, browsers will treat the resource as if it was obtained from the domain specified in the *name* attribute of the `CNAME` record. For example, in Figure 1 the website at `bank.com` embeds an image from `omns.bank.com` (e.g., as `<img src=''https://omns.bank.com/logo.gif''>` tag). The DNS server for `bank.com`, however, contains a `CNAME` record for the *name* `omns.bank.com` which aliases to the *value* `b.omtrdc.com`. Thus, the browser will fetch `logo.gif` from the host `b.omtrdc.com` but treat the response for the purposes of the same- origin policy and cookie access control as if the image were fetched from `omns.bank.com`.

Note that Content Delivery Networks (CDNs) use the same mechanism to provide their services. A CDN customer simply registers a `CNAME` for the domain under which their website should be available and points the *value* at the DNS infrastructure of the CDN provider. The CDN's DNS server will then (e.g., based on the location of a visiting client) respond with the IP of an edge server that is "closest" to the client.

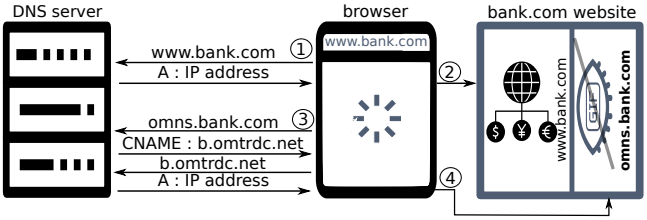


Figure 1: Sample DNS resolution example

2.2 Third-party tracking and advertising

Third-party T/A services often aggregate information (so-called user profiles) about web users and their activity across multiple websites or subdomains of a website. Since such profiles may contain personal and sensitive information, users often rely on blacklist-based browser extensions such as AdBlock [1] and Ghostery [8] to protect their privacy. These extensions prevent the browser from establishing connections with the blacklisted third-party domains. Moreover, in a push for privacy, Firefox and Safari recently started to block all third-party cookies by default.

CNAME Cloaking. To be able to continue to provide their services, T/A providers started to pivot to CNAME Cloaking as a mechanism to circumvent third-party cookie restrictions and privacy-preserving browser extensions. To this end, customers (i.e., websites) of T/A services are asked to establish a `CNAME` record in their DNS configuration that aliases a subdomain of the first-party to the infrastructure of the T/A service. Throughout this paper we refer to the established subdomain (i.e., the *name* attribute of the `CNAME` resource record, cf. `omns.bank.com` in Figure 1) as the *cloaking domain*, and the target of the alias (i.e., the *value* attribute of the `CNAME` resource record, cf. `b.omtrdc.com` in Figure 1) as the *cloaked domain*. This setup allows T/A services to deliver their resources to browsers thereby side-stepping any privacy-enhancing extensions that would prevent communication with the cloaked domain [2], and cookies set from these resources are considered first-party by the browser.

2.3 Threat Model

Scenario Setup. In our threat model, a victim interacts with an HTTPS website that cloaks a third-party T/A service as shown in Figure 2. As a first step, the victim successfully establishes an authenticated session on the website (e.g., by logging into their account). To maintain the session, the website replies by setting session cookies. As is common practice, the website owner sets the cookies' `Domain` attribute to the domain of the website. The user's

[2]Note that recent versions of Firefox make the whole DNS resolution process available to extensions, allowing them to detect and block T/A services from abusing this CNAME cloaking mechanism.

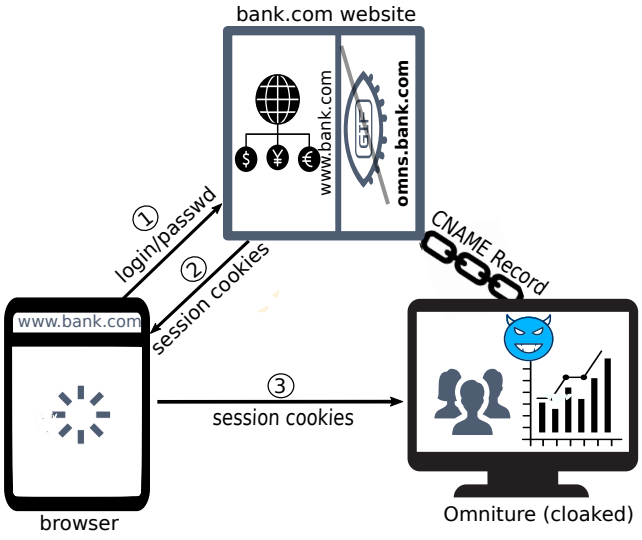


Figure 2: Cookie Leakage Scenario

browser then shares these cookies with all subdomains under the website's umbrella, including any cloaked third-parties.

Attacker's Capabilities. The "attacker" in our scenario is a T/A service that aims to hijack user sessions. That is, they seek to impersonate the user by seizing control of the user's active session using session cookies. Pursuant to business practices, our adversarial T/A service *convinces* a website to create CNAMEs pointing to its infrastructure. In our scenario, this implies that the T/A service receives qualifying first-party cookies (i.e. cookies where the Domain attribute represents a substring of the cloaking domain). These cookies are sent as a part of the HTTPS requests from the user's browser.

We acknowledge that website owners may willingly share user data and cookies with a T/A service for better targeting. It is also highly unlikely that a T/A service would perform such attacks in an official capacity. However, an administrator at the T/A has sufficient access to server logs which enables him to independently hijack user sessions. As it is entirely unnecessary for T/A services to receive session cookies in the bundle with the rest of first-party cookies, we argue that this security threat deserves attention and should be addressed promptly.

3 APPROACH OVERVIEW

In this section, we provide a high-level overview of our system (TAFinder) to assess the leakage of cookies to cloaked third parties. Specifically, to identify whether cookie leakage happens on a given website $\mathcal{W}$, TAFinder has to answer three questions: (i) Does $\mathcal{W}$ contain resources loaded from domains $\mathbb{C}_{\mathcal{W}}$ that are CNAMEs to third-parties? (ii) As explained in Section 2, CNAMEs have broader applicability than T/A services alone (e.g., for CDNs). Thus, TAFinder has to establish for each domain $\mathcal{D} \in \mathbb{C}_{\mathcal{W}}$ whether $\mathcal{D}$ points a T/A service or not (e.g., if it is a CDN). (iii) Finally, TAFinder has to assess whether $\mathcal{W}$ sets cookies with lax access control manifested through `Domain` or `Secure` attributes. If TAFinder can assert

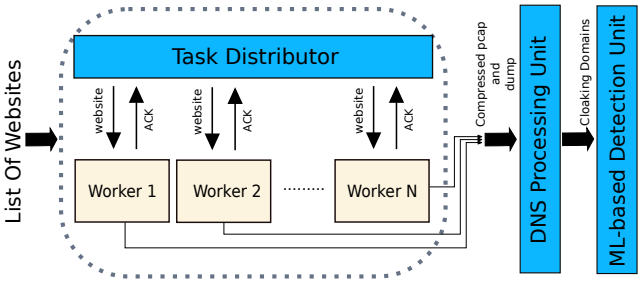


Figure 3: TAFinder Overview

all three aspects discussed above, visiting $\mathcal{W}$ with a browser will result in the automatic transmission of cookies to the T/A service.

3.1 Analysis Flow

Identify cloaking domains. Recall from Section 2 that a domain is cloaked if a DNS CNAME entry's *name* and *value* attributes are in different DNS zones (frequently different second-level domains). Thus to identify whether a website $\mathcal{W}$ includes resources from cloaked domains, TAFinder visits $\mathcal{W}$ in a browser and records all DNS traffic that results from this visit. Subsequently, TAFinder analyzes this DNS traffic and any DNS resolution that involves a CNAME record where the `Name` belongs to the domain of $\mathcal{W}$ and the *value* belongs to a different domain, is considered a cloaking domain and hence added to the set $\mathbb{C}_{\mathcal{W}}$.

Distinguish T/A services from other CNAME uses. TAFinder first labels cloaking domains in $\mathbb{C}_{\mathcal{W}}$ as T/As if they or their cloaked counterparts are present in widely used blacklists (e.g., EasyList) or categorized as such by Virus Total. However, in all likelihood, this categorization is incomplete and misses domains that are T/A services. Moreover, unlabeled domains in $\mathbb{C}_{\mathcal{W}}$ may also include CDNs that serve T/A content.

To detect T/A services among the remaining domains in $\mathbb{C}_{\mathcal{W}}$, TAFinder deploys supervised machine learning. In this case, the system examines only the cloaking domains instead of the cloaked domains since it allows to distinguish cases when T/As deliver their services through CDNs. TAFinder classifies previously unlabeled cloaking domains in $\mathbb{C}_{\mathcal{W}}$ as T/A or non-T/A by extracting features from the characteristics of the services provided by T/A domains.

Identify lax access control on cookies. TAFinder extracts cookie attributes from the HTTP traffic generated when loading a website $\mathcal{W}$. It specifically monitors $\mathcal{W}$'s cookies where `Domain` attribute is set to any ancestor-domain of $\mathcal{W}$'s host that sent the cookies. Subsequently, TAFinder verifies cookie leakage iff these cookies are included in the HTTP requests to the cloaked T/A domains. Finally, the system also tracks `Secure` cookies sent by $\mathcal{W}$.

3.2 TAFinder

Our data collection system consists of the Task Distributor module, several Workers, DNS processing, and ML-based Detection Units as shown in Figure 3.

Data to Collect. TAFinder captures DNS and HTTP(S) requests and responses sent or received by the browser when loading $\mathcal{W}$

and its resources. To examine the traffic, we record all HTTP communications in plaintext.

3.2.1 Data Collection Design Overview. TAFinder is designed in a pipeline fashion. It accepts a list of websites as input which is then distributed by the Task Distributor among Workers. Upon receiving a new $\mathcal{W}$ from the Task Distributor, a Worker spawns a new instance of a crawler. The crawler visits the $\mathcal{W}$ while logging both network packets and plaintext HTTP requests/responses and eventually transfers the captured data to the DNS Processing Unit. Notably, Workers run in separate containers, as it allows TAFinder to easily separate the network traffic from different websites.

3.2.2 DNS Processing Unit. TAFinder extracts a set of cloaking domains $\mathbb{C}_{\mathcal{W}}$ from any given $\mathcal{W}$ as described in Section 3.1. To recap, DNS Processing Unit traverses DNS resolution chains for each subdomain of a $\mathcal{W}$ that involves CNAME aliasing. If a resolution chain ends with a CNAME that belongs to a third-party, the DNS Processing Unit marks the corresponding subdomain as a cloaking domain $\mathcal{D}$.

3.2.3 ML-based Detection Unit. TAFinder starts by identifying domains in $\mathbb{C}_{\mathcal{W}}$ as T/A services using blacklists and Virus Total. To classify the remaining unlabeled cloaking domains in $\mathbb{C}_{\mathcal{W}}$ as T/A or non-T/A, TAFinder deploys a supervised machine learning approach. To this end, we engineered a set of nine features that capture the behavioral patterns inherent to T/A services as opposed to other domains. The features are extracted from the HTTP communications with the cloaking domains that arise when visiting $\mathcal{W}$. In essence, the classifier has to decide for each cloaking domain $\mathcal{D} \in \mathbb{C}_{\mathcal{W}}$ whether it is a T/A service or not. We describe our nine features below.

Feature 1: Number of HTTP requests destined to $\mathcal{D}$ over the number of all HTTP requests when loading $\mathcal{W}$ with all included resources. TAFinder's first feature originates from the observation that cloaked CDNs or other non-T/A domains aim at serving the website's content to a user. Thus, the data transferred from these domains involves a high number of HTTP requests from the user's browser. On the contrary, websites embed a limited number of resources (e.g., beacons) belonging to the T/A domains, which, in turn, requires only a limited number of the HTTP requests to be generated.

Feature 2: Total size of HTTP responses from $\mathcal{D}$ over the sum of all HTTP response sizes when loading $\mathcal{W}$ with all included resources. T/A domains supply web resources that may vary in size, but cumulatively constitute a small part of all data received during a website visit. Therefore, similar to the first feature, TAFinder compares the total sizes of HTTP responses from $\mathcal{D}$ to the size of all received HTTP responses.

To illustrate the distribution of HTTP request/response sizes across cloaked CDNs that serve website content and T/As, we examined the HTTP traffic generated from visiting *cnn.com*, the most popular website that cloaks T/As. As seen in Figure 4, the number of requests targeting the CDNs is significantly higher than of those directed to the T/A domains. Similarly, the content delivered by the CDNs is significantly bigger in size than the content supplied by the T/As.

Feature 3: Total number of cookies set by $\mathcal{D}$. TAFinder integrates this feature since T/A domains often store various user information

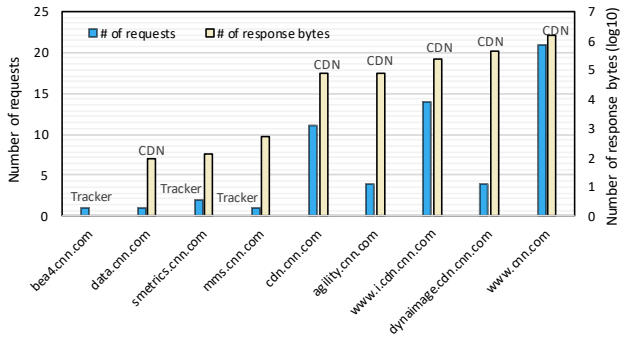


Figure 4: Distribution of requests and response sizes across cloaking domains for *cnn.com*

in cookies. Moreover, as shown in previous work by Cozza et.al. [28] the number of cookies set T/As can be significantly larger in comparison with other domains.

Feature 4: Number of long cookies over the number of all cookies set by $\mathcal{D}$. According to TrackAdvisor [44], the number of cookies where the length of the value field is higher than 35 characters is also a distinctive feature of T/A domains. Hence, TAFinder extracts the ratio of long cookies over all cookies observed in the HTTP responses from $\mathcal{D}$.

Feature 5: Number of traditionally non-targeting cookies set by $\mathcal{D}$. Regular websites are often powered by various web platforms and frameworks that require cookies to operate correctly. For example, websites created using Asp.Net, frequently use cookies called *ASPSESSIONID* to maintain a user's session. We refer to such cookies as *default cookies*.

TAFinder's feature is based on the observation that these platforms and frameworks use recognizable cookie names which may allow it to identify websites with non-tracking functionality. To this end, we compiled a set of default cookies that are most frequently sent by non-T/A domains as described in detail in Section 4.2. TAFinder then counts how many times $\mathcal{D}$ sends cookies that belong to the set.

Feature 6: Number of traditionally targeting cookies set by $\mathcal{D}$. T/As use distinct cookies to store various information about a user across multiple domains. For instance, a cookie named *SIDCC* carries information about a user's browser history [5]. Therefore, similar to the previous feature, TAFinder establishes a set of cookies (more details in Section 4.2) that are traditionally set by T/A domains and counts how many of these cookies appear in the HTTP responses from $\mathcal{D}$.

Feature 7: Number of parameters used in a URL query string over the number of HTTP requests destined to $\mathcal{D}$. Previous work [28] shows that the number of URL parameters in the request URL is an indicator of a $\mathcal{D}$ pointing to T/A provider. TAFinder incorporates this signal into its classifier as the average number of parameters per HTTP request to $\mathcal{D}$.

Feature 8: Number of times when a URL query string to $\mathcal{D}$ includes $\mathcal{W}$'s domain as one of the parameter values. This feature is based on the observation that HTTP requests targeting T/As often contain information (e.g., the domain name and currently loaded webpage)

about $\mathcal{W}$. Frequently, this information is transferred as a part of the URL query string, in the form of a parameter value. Therefore, TAFinder introduces a feature that indicates how many times the domain of $\mathcal{W}$ occurs in URL query strings submitted to $\mathcal{D}$.

Feature 9: The Content-Type feature. T/As often send invisible images, Javascripts, etc., as shown by Fouad et.al. [33]. Following this finding, we compiled a list of the corresponding HTTP content types that resemble those distributed by the T/A services as described in Section 4.2. TAFinder embeds this feature as a number (0) and increments it depending on whether the content type in an HTTP response from $\mathcal{D}$ belongs to the list of pre-compiled content types.

We use these features and train a random forest classifier as explained in more detail in Section 4.2.

4 IMPLEMENTATION

In this section, we describe the implementation of TAFinder. We first briefly discuss Data Collection and DNS Processing Units and then provide relevant data analysis details and decisions for the ML-based Detection Unit of the system.

4.1 Data Collection and DNS Processing

We realize TAFinder's first two Units in the following way. To implement the Task Distributor we leverage RabbitMQ [15], an open-source message-broker software, that allowed us to scale and run ten Workers in parallel. The Workers, in turn, are represented by the independently running Linux containers that include Tcpdump [22] and MitmProxy [13] (both in default settings)to log both network packets and pain text HTTP requests/responses. Each Worker uses a crawler powered by Selenium [20], an automated testing tool for web applications, to start a browser instance (Firefox 71.0) for each website. After a 15 seconds timeout, the crawler destroys the browser instance, compresses, and transfers the captured data to central storage. Finally, the DNS Processing Unit extracts cloaking domains along with the corresponding cloaked third-party domains for each website using Python's Scapy library [17].

4.2 ML-based Detection Unit

Domain Labeling. To label the extracted cloaked domains, TAFinder relies on multiple sources comprising commonly featured blacklists and Virus Total [24]. More specifically, TAFinder first deploys notracking's community-curated blacklist [9] which integrates a variety of blacklist sources (e.g., EasyList[7] and EasyPrivacy[6]). Based on this list, TAFinder labels known T/As. Furthermore, TAFinder labels cloaked domains that are themselves listed in the Majestic Million dataset but not present on the blacklist, as non-T/A. The intuition behind this decision is that popular T/A domains would be included in the blacklist already, and hence flagged appropriately.

To label the cloaked domains in the dataset that were not covered by either list, TAFinder relies on the categories provided Virus Total. However, Virus Total consolidates information from several different sources. As a result, there is no uniform category name that describes T/A domains. For example, *opentracker.net*, a domain that provides T/A services, belongs to '*premium*', '*computer and software*', '*web analytics*', and '*computers internet, business economy*' categories simultaneously. Thus, we empirically established several category names which Virus Total may use to characterize T/As including "*web analytics*", "*web and email marketing*", "*ads*" and "*advertisements*". It is worth mentioning that both T/As and CDNs are frequently marked as "*information technology*". Therefore, TAFinder only labels a domain in the "*information technology*" category as T/A if it simultaneously is in the "*marketing*" category.

Feature Extraction. TAFinder extracts all features from Mitmproxy's captures that contain plain text HTTP communications. The system parses this data using Mitmproxy's Python module [13]. Two of TAFinder's features reflect on the presence of traditionally targeting and non-targeting cookies. Given a large pool of distinct cookie names to classify, TAFinder only uses the most popular cookies. To this end, we extracted the names of the cookies contained in HTTP responses from all cloaking domains. We then ranked the cookies by the number of cloaking domains that sent them. In the same way, we created two more popularity rankings that are specific to cookies set by non-T/A and T/A services.

We used the most popular 25 cookies from the general cookie ranking derived above and 60 cookie names affiliated with the T/A and non-T/A domains. We categorized these cookies as targeting/advertising cookies in accordance with One Trust's cookie classification that is accessible via Cookiepedia [5] or T/A's own description, when possible. Based on this approach, we found 27 traditionally tracking-related and 28 default web application cookies featured in Table 3 in the Appendix.

In addition, TAFinder also uses a set of 24 pre-compiled content types for Feature 9 that is provided in Table 4 in the Appendix. Content types included in this set match the filetypes of resources that are often supplied by T/A services according to the prior work [33]. We manually derived this set from IANA's Common MIME Types list [4].

Classifier. To classify the cloaking domains TAFinder uses Scikit Learn's [18] random forest implementation.

5 EVALUATION

In this section, we present the results acquired using TAFinder and we discuss the composition of the CNAME cloaking ecosystem. Moreover, we describe the classification results of TAFinder for previously unknown cloaking domains. Finally, we demonstrate how the inclusion of CNAME cloaking combined with the lax access control settings for cookies impacts the security of web services.

Table 1: Distribution of the cloaked and cloaking domains

Category/#	# of Cloaked Domains	# of Cloaking Domains	# of Websites
T/A	78	2756	2,271
Others	2,062	29,656	20,504
Total distinct	2,140	32,412	21,184[3]

We evaluate TAFinder on the 100,000 most popular websites according to the Majestic Million [12] dataset. We run our experiments in January 2020 and found 93.6 % of our 100,000 dataset to

[3]Is not the sum of the column, because a single website can cloak multiple third-parties at the same time.

be reachable [4]. A detailed breakdown of these results is provided in Table 5 in the Appendix.

5.1 The ecosystem of the cloaked T/A services

Here we provide the overall statistics when evaluating our system on 100,000 most popular websites.

Websites and web analytics. 21.2 % (21,184) of the websites in our dataset use CNAME DNS records to alias at least one third-party domain. Based on the ground-truth labels obtained from the blacklist and Virus Total (see Section 4.2), ≈10.7 % (2,271) websites cloak one or more T/A services. Surprisingly, Virus Total often categorizes the websites that cloak these services as *business* and *business and economy*. This finding is concerning as websites in these categories often contain both a user's financial and private information. We also found that use of cloaked T/A services heavily skews towards popular websites as the most popular 10,000 websites disguise the highest number of T/A services (see Figure 5) which implies significant cookie leakage risks for a large number of users.

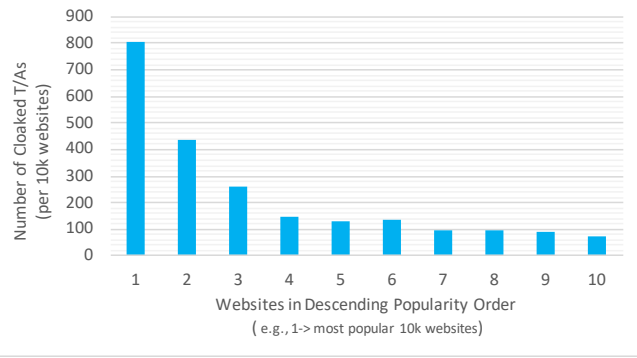


Figure 5: Distribution of T/A domains across websites (by popularity)

Cloaked Domains. Table 1 shows that 20,504 out of 21,184 websites that use CNAME aliasing integrate domains that provide non-T/A services. According to Virus Total, these non-T/As are mostly in the *information technology*, *business* and *computerandsoftware* categories which include CDN and web hosting services. Among them, `akamaiedge.net` is the most popular CDN as shown in Figure 6b (see Appendix).

The remaining 78 out of 2,140 cloaked domains represent T/A services. These domains are not equally distributed among the websites as shown in Figure 6a in the Appendix. For example, Omniture alone tracks user activities across 33.9 % (829 out of 2,271) of the websites. This result confirms the trend presented in prior work [27, 30] which also shows that a small number of trackers provide subdomain tracking services for the majority of the web.

[4]We also excluded websites that redirect their visitors to third-party domains from our analysis.

5.2 Classification results for cloaking domains

After the labeling procedure detailed in Section 4.2, TAFinder produced an imbalanced dataset that consists of 29,656 domains pointing to non-T/A hosts and 2756 cloaking domains that mask T/A services as shown in Table 1. Given the high prevalence of non-T/A, TAFinder uses a random forest (RF) classifier which performs well in the unbalanced datasets as shown in a prior work [37]. We assessed the performance of TAFinder's classifier through 10-fold cross-validation and report the classification metrics for TAFinder in Table 6 in the Appendix. Since our dataset is imbalanced, we include the macro-averaged F1 score that assigns equal weight to each class, thereby emphasizing well on rare classes (i.e., cloaking domains) as suggested by Narasimhan et.al. [32].

5.2.1 Newly identified cloaking domains. Overall, using blacklists and Virus Total, TAFinder discovered 2,128 subdomains that point to T/A services. The system also classified 29,363 cloaking domains which were not assigned labels based on the procedure described in Section 4.2. Among them, we label 20,910 domains that start with 'World Wide Web' (i.e.,'www.') as non-T/A as the homepage of every site will not be hosted on T/A servers [5]. Out of the remaining 8,453 domains, the classifier labeled 123 as T/A. To verify the classification results, we manually analyze the HTTP communications with randomly selected 27 out of 123 domains.

More specifically, we examined the contents fetched from these domains and manually compared any JavaScript resource with those from known T/A services. To this end, we compared functions, variable names, and comments. Furthermore, we examined cookies and the static (i.e., non-script) content received. In this way, we discovered 11 domains that exhibit clear tracking behavior. As a prominent example, we found that a website that belongs to TD bank tracks its visitors through `smetrics.td.com` (hosted on `taucdn.net`). Similarly, CDW Corporation, a leading technology solutions provider, also deploys Omniture's analytics services disguised as `smetrics.cdw.com` (points to `akamaiedge.net`). While we were unable to check if TD bank leaks session cookies to the embedded trackers (i.e., we had no customer accounts with the bank), we found that at least regular cookies that store user preferences are leaked to its' analytics provider (i.e., Omniture). In the case of CDW, we were able to register an account. We verified that session cookies were indeed shared with the analytics provider following the procedure described in Section 5.3.

Out of the remaining 16 cloaking domains, 3 were falsely classified. We found that requests to the four domains were either generated using jQuery scripts that are often used for dynamic content generation or simply supply statically embedded images. Finally, 4 results were inconclusive (e.g., the corresponding JavaScript files were either highly obfuscated) and 9 subdomains and websites no longer existed.

5.2.2 Feature Sensitivity. We also performed Recursive Feature Elimination (RFE) analysis [16] that allows us to evaluate the relative contributions, i.e., importance, of the features in our classifier. RFE calculates the importance of each feature by removing one

[5]The Firefox browser that we used in our experiments prepends 'www.' to complete an otherwise naked domain [14, 19]. Therefore, every URL that starts with 'www.' is likely to point to a homepage of a website.

feature at a time and recalculating the accuracy of our classifier. As shown in Figure 7, the two features with the most discriminative power are Feature 2 (ratio of the HTTP response sizes from a cloaking domain) and Feature 7 (average number of URL query parameters). This correlates with our expectation of T/A services that do not supply any of a website's content. At the same time, the trackers require meta-information about a page where the HTTP requests originate from. In the current realm where third-party cookies are or about to be blocked by the major browsers ([36] [35], [38]), T/A providers often rely on URL parameters. Notably, although we only included 27 traditionally targeting cookie names as one of our features (Feature 6), its significance is higher than the significance of the long cookies feature (Feature 4), used in previous work [28], thereby emphasizing its potential.

5.3 Leaking Cookies

In our threat model, cloaked T/A domains receive first-party cookies via HTTP requests if the first-party explicitly shares the cookies with all its subdomains. We observed that 1,195 out of 2,271 websites that cloak T/A services, leak the first-party cookies to one or more T/A domains. Based on the fact that we have evidence of the above cookie leakage from web browsing sessions, these results are a lower-bound estimate. One reason why the true number of affected cookie-leaking websites can be higher is the impact of the European General Data Privacy Regulation (GDPR). The GDPR requires user-consent before any cookies can be sent to a user's browser. Since our crawler does not interact with any opt-in mechanisms to accept cookies, a GDPR-compliant site would not send any cookies to the crawler and TAFinder cannot analyze whether these cookies would be leaked. However, we also do not expect the number of websites that leak their cookies to be drastically higher either: Sanchez-Rola et.al. [34] discovered that only 2.5% of 2,000 most popular websites adhere to a user's tracking opt-out preference.

The impact of cookie leakage is two-fold. First, the T/A services may acquire session identifiers, authentication tokens, and other sensitive information. Second, we found that 437 out of 2,271 websites in our dataset communicated with cloaked T/As using plain text HTTP. Seven websites among these set the `Secure` attribute for their cookies, thus ensuring the cookies do not get exposed on the plain text HTTP channel. However, the remaining 430 sites do not set the `Secure` attribute and hence expose their users' accounts to an additional threat posed by an on-path network attacker. To make matters worse, this leak and vulnerability even exists if the entire first-party site forces HTTPS [39, 43].

5.3.1 Leaked Cookies and Online Accounts. The above discussion considers a website as vulnerable if any of its cookies are leaked to a cloaked T/A. However, it is possible that websites configure their session cookies with appropriate access restrictions via the `Domain` attribute. Thus, to investigate how cookie leakage affects the security of the online user accounts we check whether websites leak *session* cookies to the T/As. This, however, is not a trivial task. There is no comprehensive way to identify cookies that carry session identifiers. Moreover, establishing an authenticated session with a website requires logging in or signing up — a highly diverse activity considering that the majority of the websites in our dataset do not use common authentication schemes. As such, we were only able to partially automate this task.

For this assessment, we compiled a set of websites that cloak a T/A domain and provide "*banking*", "*financ*", or "*shopping*" services according to Virus Total. We consider these categories to be the most security-sensitive since accounts on such websites usually collect financial and personal information including billing and shipping addresses, credit card information, purchase histories, etc.

We then examined whether the cookies shared with the T/A services allow access to the user accounts, i.e. contain *session* cookies. For every website in our list, we either used an existing account or created a new one to establish an authenticated session when possible. We were not able to acquire accounts with several banks and payment agencies that require financial and personal information (e.g., a Social Security Number) to register. In this experiment, we tested every site where we successfully obtained online accounts as follows. We first manually logged in into our account, recorded a string that is only shown for an authenticated user (e.g., 'Account Balance') and passed it to a crawling script. The script then extracted the cookies that were shared with all subdomains of the tested site during our session. As the next step, in a separate browser instance running on a device with a different IP address, the script navigated to the site and injected the collected cookies. Using the previously recorded string the script finally checked whether our account could successfully be accessed in the second browser instance. In that case, we marked the tested website as vulnerable since it leaked session cookies along with others first-party cookies to a T/A service.

Since this procedure requires manual effort for each analyzed site, we limited the experiment to 10,000 most popular domains. After filtering financial service and shopping websites, we ended up with 119 domains to analyze. We were able to create accounts on 90 of these sites. Concerningly, 27 (see Table 2) of these 90 websites leak their session cookies, which we confirmed to provide access to the user's account, to cloaked T/A services. The majority of the affected websites are online shopping retailers which often provide their authenticated users with the ability to manage or save their store or regular credit cards for further purchases. The situation is particularly dire in the case of `walgreens.com` where a malicious actor was able to obtain the refill prescription and order history of the user. Note that Walgreens fixed this vulnerability after we informed them. However, given that they never responded to our reports, we cannot conclusively claim that the fix was deployed as a consequence of our reporting. Finally, we found that two major US banks also expose session cookies to the T/As, granting them the capability to make payments and obtain the account balance, rewards, and other financially sensitive information. As mentioned in the discussion on the threat model, while it is unlikely that well established T/A providers such as Adobe or TechSolutions will attempt to illegitimately access user accounts, the accounts still remain at risk. This is due to the fact that anyone (e.g., disgruntled or ill-intended T/A administrator) with access to request logs may takeover the accounts.

Session Respring Vulnerabilities. If implemented properly, server-side session management should invalidate the session if the authenticated user logs out of the service. Failure to do so opens up

Table 2: Vulnerable vendors from Majestic top 10,000 sites

Domain Name	Majestic Ranking	Session Hijacking	Session Respring
walmart.com	478	yes	yes
nike.com	551	yes	yes
sky.com	602	yes	no
bestbuy.com	1,014	yes	no
homedepot.com	1,210	yes	no
newegg.com	1,546	yes	no
walgreens.com	2,305	yes	no
overstock.com	2,539	yes	yes
victoriassecret.com	3,306	yes	yes
staples.com	3,339	yes	yes
fnac.com	3,694	yes	no
fandango.com	3,953	yes	yes
jcpenney.com	4,006	yes	no
moodys.com	4,690	yes	yes
chegg.com	4,750	yes	no
kroger.com	5,538	yes	no
indigo.ca	6,097	yes	yes
realestate.com.au	6,921	yes	yes
hallmark.com	7,194	yes	no
carfax.com	7,501	yes	no
harborfreight.com	7,766	yes	no
westelm.com	7,967	yes	no
landsend.com	8,615	yes	no
stubhub.com	8,876	yes	no
bestbuy.ca	9,664	yes	no
bank1.com (undisclosed)	(undisclosed)	yes	yes
bank2.com (undisclosed)	(undisclosed)	yes	no

so-called session respiring attacks where the attacker replays stale session cookies to revive a session that should have been terminated. In this evaluation, we assessed the 27 sites that leak session cookies for their vulnerability to session respring attacks. Adding insult to injury, we discovered that 10 (37 %) websites that leak session cookies are also vulnerable to session respring attacks, exposing their users' accounts to increased risk of take over.

Responsible Disclosure. We disclosed our findings to all of the affected vendors in the beginning of April 2020. Unfortunately, most merely provide generic contact forms or regular customer service contacts. As such, to date, we only received six responses. Among them is the response from the team that manages a bug bounty program for one of the US largest banks. [6]. The message expressed indifference towards violating the confidentiality of the session cookies by exposing them to third-party domains. According to the team, third-party Javascript served from the cloaking domain would be treated as first-party code and hence has full access to all first-party cookies. However, this argument is fallacious since session cookies can be protected from Javascript access via the `HTTPOnly` attribute. Unfortunately, several other vendors, including *walgreens.com*, have not replied to us, but patched the vulnerabilities after we informed them.

6 DISCUSSION

In this section, we briefly discuss the limitations of our work and potential mitigations against cookie leakage due to CNAME cloaking and lax cookie access control.

[6]Unfortunately, the conditions of the bug bounty program prevent us from publicly disclosing the identity of the bank.

Limitations. Our work has several limitations. The scope of our CNAME cloaking analysis is limited to the third-parties included in the first-party's homepage. It also excludes cases where websites sublease one of their subdomains (e.g., *coupons.cnn.com*). Moreover, TAFinder monitors cookies received via HTTP responses and cannot determine the exact origin of the cookies set from Javascript. The system also uses the default settings for Tcpdump ring buffer which may result in the packets loss. Therefore, the number of first-party domains whose cookies are potentially leaked to T/As may actually be higher than what we reported, as well as the number of the domains that deploy CNAME cloaking.

Finally, our system uses community-based blacklists to label domains in our dataset, making its accuracy depend on the accuracy of the blacklists.

Mitigation mechanisms. To mitigate the security implications of CNAME cloaking, website owners can simply adjust the cookie attributes. In particular, websites can restrict the scope of the first-party cookies by omitting the `Domain` attribute, in which case cookies will only be shared with the precise host that sets them. Moreover, websites should use the `Secure` attribute, which will force user browsers to include the cookies only over HTTPS connections. Finally, if websites execute third-party Javascript in the first-party context, they should set their cookies as `HTTPOnly` to prevent access to the first-party cookies.

Another approach to handle security concerns related to CNAME cloaking is to populate blacklists deployed by browser extensions, such as AdBlock [1] with a list of cloaking domains. However, a more robust mechanism would allow such extensions to tap into the DNS resolution process to identify cloaked domains directly. Currently, this is only possible for extensions on the Firefox browser, and uBlock Origin [23] already makes use of this functionality. Once other browsers follow suit, the privacy-enhancing extensions on these platforms will also be able to block cloaked domains and restore user security and privacy. However, extensions are always bandaid solutions and we hence advocate for the more robust solution of properly configuring access control of session cookies by the first-party website owner.

7 RELATED WORK

In this section, we briefly summarize existing research on CNAME cloaking as well as online tracking. We then discuss relevant cookie leakage scenarios covered presented in the prior work.

7.0.1 CNAME cloaking and other web tracking studies. To the best of our knowledge, the first work that reported the inclusion of third-party T/A domains as the first-party is dating back to 2006 [25]. The paper analyzes how information about a user is gathered across a wide variety of websites including both visible, i.e. first-party and *hidden*, i.e. cloaked domains. In a follow-up paper in 2009 [27], Krishnamurthy et.al. described a longitudinal study of the 1200 most popular websites with respect to the embedded third-party domains. The results of the study showed the growth in the use of the CNAME cloaking technique by 30% in the span of 3 years. Another work conducted by Wills et.al. [29] found the presence of 316 cloaked T/A domains in at least 1% of popular websites. Wills

et.al. [29] also discuss the limitations of the T/A blocking browser extensions from the perspective of CNAME cloaking.

Several works have also been dedicated to measuring tracking on a large sample of the most popular websites. Limbert [45] analyzed HTTP requests generated by visiting 1 million Alexa websites to understand the scale and the nature of third-party domains. Later work by Englehardt et.al. [41] presented the distribution of T/As among websites, as well as the statistics on the cookie syncing performed by the T/A domains. Merzdovnik et.al. [31] examined the efficiency and effectiveness of several popular tracking defenses on 100,000 popular websites and 10,000 Android applications. The paper [31] also acknowledges the challenge of blocking tracking attempts from domains hosted on CDNs.

Dao et.al. [30] concurrently and independently conducted a measurement of the CNAME cloaking ecosystem of Alexa's top 300K websites [2]. Our study reports similar results in the general CNAME cloaking trends such as the distribution of the T/A services across websites by popularity, category (*business* and *business and economy*), etc. Since we used the Majestic Million [12] dataset that significantly differs from Alexa's list [2] as previously shown by Scheitle et.al. [40], only 31,020 websites were analyzed by both studies. Moreover, due to the difference in community-supported blacklists (varying blacklist sources and, potentially, access dates), the results of both studies cannot be directly compared. In the spirit of giving back to the community, we will make a list of newly found subdomains that disguise analytics services based on the most up-to-date blacklist[9] publicly available. We will also open-source the list of websites that we found to use CNAME cloaking in this paper.

Generally, as opposed to the privacy-only focus of prior work, we demonstrate how the combination of CNAME cloaking and misconfigured cookie access controls lead to serious security issues where the confidentiality of session cookies is 'destroyed'.

7.0.2 *Cookie Leakage Scenarios.*

via CNAME cloaking. Krishnamurthy et.al. [26] described the potential privacy risks linked to Online Social Networks (OSNs) embedding third-party T/A domains. In particular, the authors described how personally identifiable information stored with the OSNs can be leaked to the trackers through CNAME cloaking. Krishnamurthy et.al. [26] also observed that 2 out of 12 OSNs leak the OSN id to the Omniture domain. In our paper, we expand the analysis of CNAME cloaking deployment to a large number of websites and reveal the leakage of session cookies to T/A providers.

via HTTP/ mixed content. Several works focus on the cookie exposure due to the use of unencrypted HTTP protocol [39], [42], [43]. In particular, Englehardt et.al. [42] examine cases when websites connect to T/A domains over HTTP. The paper concluded that by passively eavesdropping on network traffic, an adversary may profile users with higher accuracy than conventional methods such as by comparing IP addresses. Sivakorn et.al. [43] found that 284K user accounts in their 30 days experiment are susceptible to HTTP cookie hijacking attack due to the mixed HTTP and HTTPS content. They detailed the capabilities that an adversary can obtain in major websites, and also explained how HTTP cookie exposure can be applied in deanonymizing Tor users. Similarly to these works, we are also concerned with the use of the HTTP protocol. However, in our scenario, the adversary can be represented as both a network attacker and an adversarial T/A service.

8 CONCLUSION

In the web ecosystem where websites often leverage user data to generate additional revenue or use analytics insights, online user accounts may be endangered by incorrect use of subdomain tracking. In this work, we demonstrated how using CNAME cloaking to realize the tracking along with the lax access control on first-party cookies leads to the breach in confidentiality of user session cookies (i.e., anyone with HTTP request logs can access user accounts). We also developed a system that is capable of automatically detecting the presence of cloaked T/A services, and the first-party cookie leakage. We found that ≈ 10.7 % of the most popular websites deploy subdomain tracking with nearly Half of them leaking first-party cookies to the T/As. Given the potential spread of this form of tracking, we hope that our work brings attention to the issue of the security of online accounts.

REFERENCES

[1] [n.d.]. AdBlock. https://getadblock.com/block-ads-and-popups.
[2] [n.d.]. Alexa Top Sites. http://s3.amazonaws.com/alexa-static/top-1m.csv.zip.
[3] [n.d.]. CNAME Cloaking, the dangerous disguise of third-party trackers. https://shorturl.at/biqEF.
[4] [n.d.]. Common MIME types. https://developer.mozilla.org/en-US/docs/Web/HTTP/Basics_of_HTTP/MIME_types/Common_types.
[5] [n.d.]. Cookiepedia. https://cookiepedia.co.uk.
[6] [n.d.]. *Easy Privacy*.
[7] [n.d.]. EasyList. https://easylist.to.
[8] [n.d.]. Ghostery. https://www.ghostery.com.
[9] [n.d.]. Github. https://github.com/notracking/hosts-blocklists.
[10] [n.d.]. Global Digital Population as of July 2020. https://www.statista.com/statistics/617136/digital-population-worldwide/.
[11] [n.d.]. Internet 2010 in numbers. https://www.pingdom.com/blog/internet-2010-in-numbers/.
[12] [n.d.]. Majestic Million. https://majestic.com/reports/majestic-million.
[13] [n.d.]. MitmProxy. https://github.com/mitmproxy.
[14] [n.d.]. PC Magazine ENCYCLOPEDIA. https://www.pcmag.com/encyclopedia/term/naked-domain.
[15] [n.d.]. RabbitMQ. https://www.rabbitmq.com.
[16] [n.d.]. Recursive Feature Elimination. https://www.scikit-yb.org/en/latest/api/model_selection/rfecv.html.
[17] [n.d.]. Scapy. https://scapy.net.
[18] [n.d.]. Scikit Learn. https://scikit-learn.org.
[19] [n.d.]. Search the web from the address bar. https://rb.gy/9ccb6a.
[20] [n.d.]. Selenium webdriver. https://www.selenium.dev/projects/.
[21] [n.d.]. Session Management Cheat Sheet. https://cheatsheetseries.owasp.org/cheatsheets/Session_Management_Cheat_Sheet.html.
[22] [n.d.]. Tcpdump. https://www.tcpdump.org.
[23] [n.d.]. uBlock Origin. https://github.com/gorhill/uBlock.
[24] [n.d.]. Virus Total. https://www.virustotal.com/gui/home/upload.
[25] B.Krishnamurthy and C.Wills. 2006. Generating a Privacy Footprint on the Internet. In *Proceedings of the 2006 ACM SIGCOMM Conference on Internet Measurement (IMC'06)*. 65–70. https://doi.org/10.1145/1177080.1177088
[26] B.Krishnamurthy and C.Wills. 2009. On the Leakage of Personally Identifiable Information Via Online Social Networks. In *Proceedings of the 2009 ACM workshop on Online social networks*. 112–117.
[27] B.Krishnamurthy and C.Wills. 2009. Privacy Diffusion on the Web: A Longitudinal Perspective. In *Proceedings of the 18th International Conference on World Wide Web* (Madrid, Spain) *(WWW'09)*. 541–550. https://doi.org/10.1145/1526709.1526782
[28] F. Cozza, A. Guarino, F. Isernia, D. Malandrino, A. Rapuano, R. Schiavone, and R. Zaccagnino. 2020. Hybrid and lightweight detection of third party tracking: Design, implementation, and evaluation. *Computer Networks* 167 (2020). https://doi.org/10.1016/j.comnet.2019.106993
[29] C.Wills and D.Uzunoglu. 2016. What Ad Blockers Are (and Are Not) Doing. In *2016 Fourth IEEE Workshop on Hot Topics in Web Systems and Technologies (HotWeb'16)*. 72–77. https://doi.org/10.1109/HotWeb.2016.21
[30] H. Dao, J. Mazel, , and K. Fukuda. 2020. Characterizing CNAME Cloaking-Based Tracking on the Web. *IEEE/IFIP TMA'20* (2020), 1–9.

[31] G.Merzdovnik, M.Huber, D.Buhov, N. Nikiforakis, S.Neuner, M.Schmiedecker, and E.Weippl. 2017. Block Me If You Can: A Large-Scale Study of Tracker-Blocking Tools. In *Proceedings of the 2017 IEEE European Symposium on Security and Privacy (Euro SP'17)*. 319–333. https://doi.org/10.1109/EuroSP.2017.26

[32] H.Narasimhan, W.Pan, P.Kar, P.Protopapas, and H.Ramaswamy. 2016. Optimizing the Multiclass F-Measure via Biconcave Programming. In *Proceedings of the 2016 IEEE 16th International Conference on Data Mining (ICDM'16)*. 1101–1106. https://doi.org/10.1109/ICDM.2016.0143

[33] I.Fouad, N.Bielova, A.Legout, , and N.Sarafijanovic-Djukic. 2020. *Proceedings on Privacy Enhancing Technologies* 2020, 2 (2020), 499–518.

[34] I.Sanchez-Rola, M.Dell'Amico, P. Kotzias, D.Balzarotti, L.Bilge, P.Vervier, and I.Santos. 2019. Can I Opt Out Yet? GDPR and the Global Illusion of Cookie Control. In *Proceedings of the 2019 ACM Asia Conference on Computer and Communications Security (Asia CCS'19)*. 340–351.

[35] J.Schuh. [n.d.]. Building a more private web: A path towards making third party cookies obsolete. https://blog.chromium.org/2020/01/building-more-private-web-path-towards.html.

[36] J.Wilander. [n.d.]. Full Third-Party Cookie Blocking and More. https://webkit.org/blog/10218/full-third-party-cookie-blocking-and-more/.

[37] A. Kharraz, W. Robertson, and E. Kirda. 2018. Surveylance: Automatically Detecting Online Survey Scams. In *Proceedings of the 2018 IEEE Symposium on Security and Privacy (SP'18)*. 70–86. https://doi.org/10.1109/SP.2018.00044

[38] M.Wood. [n.d.]. Today's Firefox Blocks Third-Party Tracking Cookies and Cryptomining by Default. https://shorturl.at/rzD35.

[39] P.Chen, N.Nikiforakis, C.Huygens, , and L.Desmet. 2015. A Dangerous Mix: Large-Scale Analysis of Mixed-Content Websites. In *Information Security*. 354–363.

[40] Q.Scheitle, O.Hohlfeld, J.Gamba, J.Jelten, T.Zimmermann, S.D.Strowes, and N.Vallina-Rodriguez. 2018. A Long Way to the Top: Significance, Structure, and Stability of Internet Top Lists. In *Proceedings of the Internet Measurement Conference 2018 (IMC'18)*. 478–493. https://doi.org/10.1145/3278532.3278574

[41] S.Englehardt and A.Narayanan. 2016. Online Tracking: A 1-Million-Site Measurement and Analysis. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security (CCS '16)*. 1388–1401. https://doi.org/10.1145/2976749.2978313

[42] S.Englehardt, D.Reisman, C.Eubank, P.Zimmerman, J.Mayer, A.Narayanan, and E.W.Felten. 2015. Cookies That Give You Away: The Surveillance Implications of Web Tracking. In *Proceedings of the 24th International Conference on World Wide Web (WWW'15)*. 289–299. https://doi.org/10.1145/2736277.2741679

[43] S.Sivakorn, I.Polakis, and A.Keromytis. 2016. The Cracked Cookie Jar: HTTP Cookie Hijacking and the Exposure of Private Information. In *Proceedings of the 2016 IEEE Symposium on Security and Privacy (SP'16)*. 724–742. https://doi.org/10.1109/SP.2016.49

[44] T.Li, H.Hang, M. Faloutsos, and P. Efstathopoulos. 2015. TrackAdvisor: Taking Back Browsing Privacy from Third-Party Trackers. In *International Conference on Passive and Active Network Measurement*. 277–289.

[45] T.Libert. 2015. Exposing the hidden web: An analysis of third-party HTTP requests on 1 million websites. (2015). arXiv:1511.00619 [cs.CR]

A APPENDIX

A.1 Implementation

Table 3: Sets of traditionally T/A and non-T/A cookies used by TAFinder for Features 6 and 5

T/A cookies	non-T/A cookies
___utmvc	ASPSESSIONID
s_ecid	ASP.NET_SessionId
SC_ANALYTICS_GLOBAL_COOKIE	JSESSIONID
pardot	PHPSESSID
AMCV_xxxxxAdobeOrg	AKA_A2
cqcid	AWSELB
s_vi	ARRAffinity
sp	sid
OAID	XSRF-TOKEN
wt_nbg_Q3	_sp
OAGEO	security_session_verify
MUID	__RequestVerificationToken
VISITOR	SERVERID
mboxSession	secure_customer_sig
mboxPC	cart_sig
thx_guid	.ASPXANONYMOUS
SM	_shopify_y
MR	_orig_referrer
ANONCHK	_landing_page
s_visit	csrftoken
_sp_v1_xxxxx	language
idrxvr_	CFTOKEN
_node	CFID
_hash	ELOQUA
TS0xxxxxx	form_key
wteid_xxxxx	access_token
cluid	_uid
	__cfduid

Table 4: List of Content-Types for Feature 9

application/javascript	application/ld+json
application/x-javascript	text/javascript
application/json-patch+json	font/otf
application/json-seq	image/png
image/bmp	application/php
application/x-csh	image/svg+xml
text/css	image/tiff
image/gif	font/ttf
text/html	image/webp
image/jpeg	font/woff
text/javascript	font/woff2
application/json	application/xhtml+xml

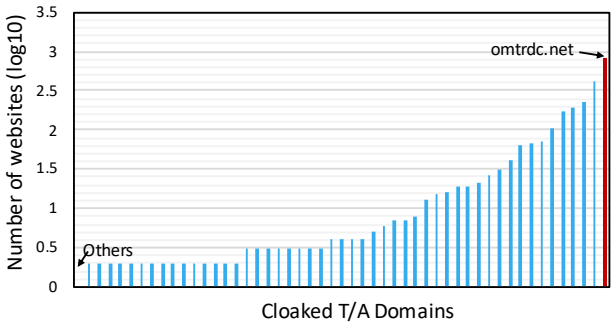
A.2 Results

Table 5: Overview of the visited 100,000 websites

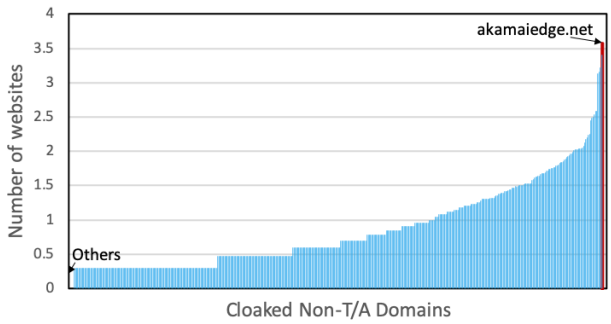
	Number of Websites/ %
No Cloaking	72,409/ 72.4 %
Cloaking a Third-Party	21,184/ 21.2 %
Unreachable	6,407/ 6.4%

Table 6: Random forest classifier metrics

Metric	Random Forest
Accuracy	96 %
Precision	95 %
F1-Score	88 %



(a) Distribution of the cloaked blacklisted T/A domains across the websites



(b) Distribution of the cloaked non-T/A domains across the websites

Figure 6: Overview of the labeled cloaked domains

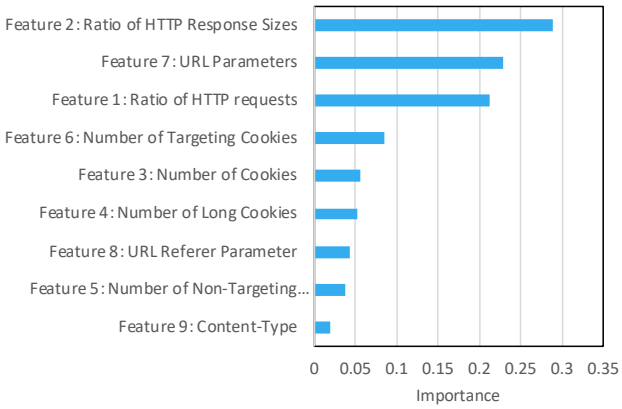


Figure 7: Feature Importances