

CDNs AND PRIVACY THREATS: A MEASUREMENT STUDY

AKASH LEVY

SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
BACHELOR OF SCIENCE IN ENGINEERING
DEPARTMENT OF ELECTRICAL ENGINEERING
PRINCETON UNIVERSITY

ADVISER: PRATEEK MITTAL

MAY 2017

I hereby declare that this Independent Work report represents my own work in accordance with University regulations.

X _____

Akash Levy

Abstract

Content Delivery Networks (CDNs) are distributed overlay networks that deliver content to end users on behalf of origin websites. They have generally been treated by both origin websites and end users as trusted entities—as a result, there has been little study on potential privacy threats that arise from their widespread usage. In this thesis, we consider privacy threats posed by CDNs that have access to a large set of user browsing information. We examine CDN usage in top websites and discuss what inferences the most popular CDNs might be able to make about end users based on the quantity and nature of the content they deliver.

Acknowledgements

I would like to thank Maria Gorlatova, Srdjan Matic, Yixin Sun, and Litian Liu for their input on CDN discovery techniques and for revisions, as well as Elad Karako, the creator of the HOSTS/Adblock domain blacklist. The figures in this thesis were made possible through FlatIcon’s free online vector graphics library.

I also must thank my close friends Zaynab Zaman, Soraya Morales, Sarah Sakha, Ambika Viswanathan, Kevin Romero, Mohamed El-Dirany, Anyssa Chebbi, Roger Van Peski, Chris Zhang, Prateek Swain, and Mimi Chung (among many others who I have surely failed to mention) for their encouragement and support throughout the writing of this thesis. I would like to thank my senior year eating club, Princeton Tower Club, and its officers/membership for giving me a sense of community in my senior year—also for the thesis room and the tastiest food I could wish for. I also must thank my parents for their endless support and their willingness to read over my thesis despite their lack of background in security/privacy domains.

I would like to thank my peers in ELE 574 who gave me ideas for improving my project as well as Prof. Arvind Narayanan and Steven Engelhardt for their excellent supporting work in privacy research. And lastly, I must profusely thank my advisor, Prof. Prateek Mittal for all of his teachings, guidance, and encouragement throughout the writing of this thesis and graduate school applications.

Contents

Abstract	iii
Acknowledgements	iv
1 Small-Scale Measurement of CDN Deployment	1
1.1 Introduction	1
1.2 Related Work	6
1.2.1 CDN measurement studies	6
1.2.2 Online privacy studies	6
1.2.3 Privacy-enhancing technologies	7
1.3 Measuring the Privacy Threat	7
1.3.1 CNAME Lookups	10
1.3.2 Reverse DNS Lookups	10
1.3.3 RDAP Record Lookups	10
1.3.4 Determining CDN Usage with Confidence	11
1.4 Validating Measurements	16
1.5 Categorical Analysis	17
1.6 Analysis of CDN Measurement	19
1.7 Conclusion	20
2 Large-Scale Measurement of CDN Deployment	22
2.1 Introduction	22

2.2	Related Work	24
2.2.1	Web privacy measurement tools	24
2.2.2	Hostname/IP to organization mapping	24
2.3	OpenWPM: Automated Large-Scale Privacy Measurement	25
2.3.1	Browser driver	26
2.3.2	Browser managers	26
2.3.3	Task manager	27
2.3.4	Data aggregator	28
2.3.5	Instrumentation	28
2.4	Measuring Third-party Content Deployment	28
2.5	Cross-Validation of Measurement Techniques	33
2.5.1	Comparison of AS/hostname/header-based techniques	33
2.5.2	Comparison of hostname-based techniques	35
2.6	Conclusion	36
3	Privacy Threats from Weak Referrer Policy	37
3.1	Introduction	37
3.2	Related Work	38
3.2.1	Studies on third-party content and privacy	38
3.2.2	Studies on traffic fingerprinting and defenses against it	39
3.3	Overview of Privacy-Leaking HTTP Headers	41
3.3.1	Referrer policy and the Referer header	41
3.3.2	Cross-Origin Resource Sharing (CORS) and the Origin header	43
3.4	Measuring Referrer Policy	44
3.5	Measuring CORS Policy	46
3.6	Reforming Referrer Policy	48
3.7	Conclusion	48

4	Resource Heterogeneity and Traffic Fingerprinting Threats	49
4.1	Introduction	49
4.2	The CommonCrawl Dataset	50
4.3	Methodology	51
4.4	Results and Analysis	52
4.5	Conclusion	52
5	Future Work and Conclusion	56
5.1	Future Projects	56
5.1.1	Deeper crawls and more intricate measurements	56
5.1.2	Monitoring third-party information leakage via real-time measurement framework	56
5.1.3	Investigating offload to multiple third-parties by origin hosts, splitting user traffic amongst them	57
5.1.4	Splitting traffic between multiple third-party options provided to end users by origin hosts	57
5.1.5	Investigating the feasibility of proxies/Onion routing to access CDN content	57
5.1.6	P2P content delivery	58
5.1.7	End users can access CDN content using multiple IP addresses	58
5.1.8	Investigating CDNi and its user privacy implications	59
5.1.9	Automated CDN whitelist generation	59
5.2	Conclusion	59
A	CNAME Whitelist for Small-Scale Measurement	61
B	Examining CDN Ownership of Tor Relays	66
	Bibliography	69

Chapter 1

Small-Scale Measurement of CDN Deployment

1.1 Introduction

Content Delivery Networks (CDNs) have gained widespread use on the Internet since the early 2000's [1]. Today, there are many commercial CDNs around the world, including Akamai, Fastly, Cloudflare, Edgecast, and Limelight. With the rise in demand for high-speed networking, CDNs have provided a way forward—they have succeeded in increasing availability of website content while reducing latency. These improvements have been possible through the use of highly distributed networks that deliver content to end users based on their geographical location [2]. This stands in contrast to traditional web content hosting, where content is hosted on a small set of co-located servers.

Figure 1.1 depicts the typical layout of a CDN. Since CDNs provide content on behalf of origin websites, trust must exist between the origin websites and the CDNs—trust has generally been *assumed* to exist between the origin websites, CDNs, and end users. In this thesis, we consider the consequences when this assumption fails

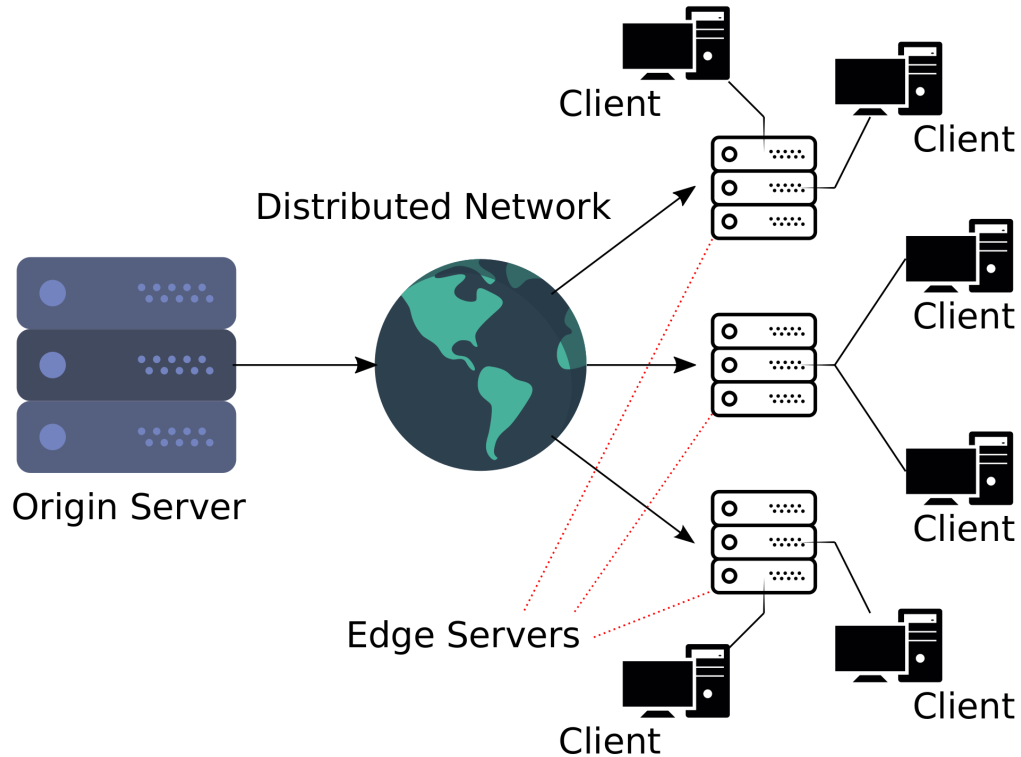


Figure 1.1: A diagram of a typical CDN. The origin server offloads its content onto a distributed network of edge servers, which serve content to end users based on their geographical location.

to hold. To the best of our knowledge, we are the first to systematically investigate privacy threats due to CDNs.

CDNs generally host static content (like images and video, as well as other web objects like JavaScript and CSS files) rather than dynamic content [1]. It is more convenient for origin websites to use CDNs for static content since this content is not updated often – as a result, replicating changes across the network is typically not time-sensitive.

Static content is also generally less sensitive than dynamic content in terms of user privacy, since it is not personalized for the end user. This is one reason CDNs are so often overlooked as threats to end users. In accessing specific static content,

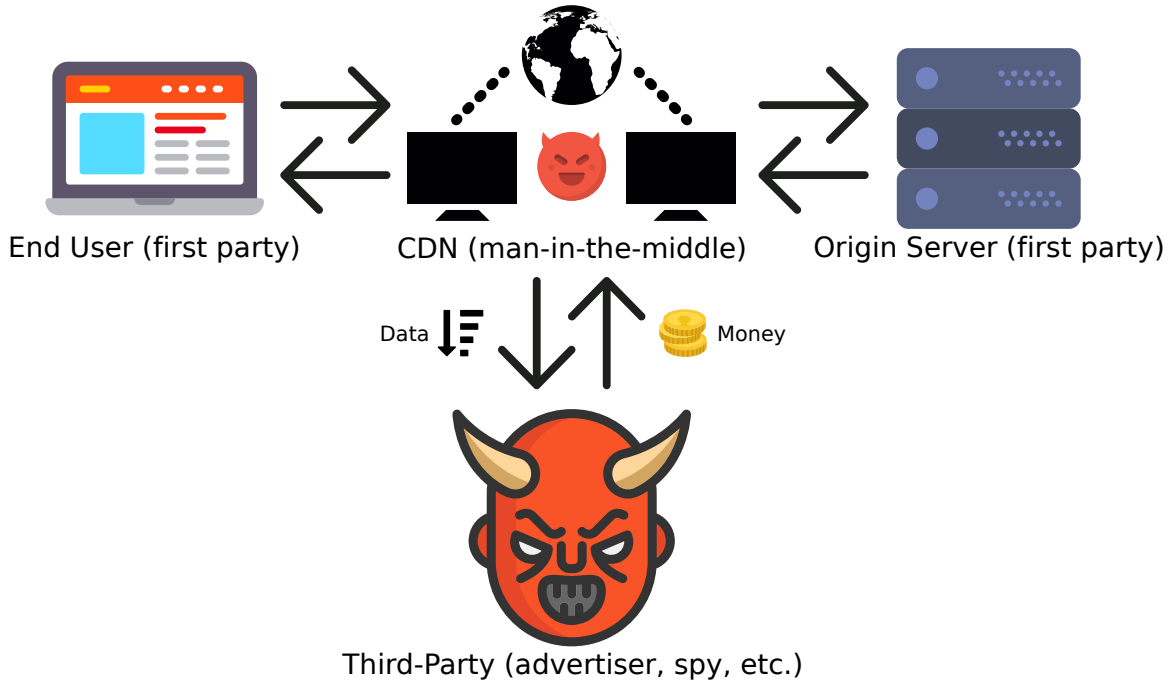


Figure 1.2: A depiction of a CDN privacy threat. The CDN acts as a man-in-the-middle, with the ability to misuse data to infer sensitive user details, and potentially collude with third-party advertisers or surveillance regimes.

however, users reveal sensitive information based on the classification of the content being accessed—for example, on a CDN-based movie streaming service, accessing particular movies could allow the CDN provider to infer movie preferences. Later, this information might be relayed to advertisers for profit, violating the privacy of the end users. More worrisome is a situation in which such information is sold to third-parties that can infer and use sensitive information about users, such as finances and political orientation. These risks persist even when strong encryption techniques such as TLS are used to secure the communications in transit, because the key exchange protocols are run by the CDN provider [3]. Government agencies may be able to buy (or seize) data for mass surveillance purposes. In 2014, documents leaked from the National Security Agency showed practices about massive collection of profile pictures from social networks through Akamai CDN endpoints [4]. Roger Dingledine, a co-founder of the Tor project [5], recently suggested that Cloudflare hosting `tips.fbi.gov` has

chilling consequences for the privacy of whistleblowers. Figure 1.2 illustrates what CDN privacy threats entail.

More recently, CDNs have even begun to deliver dynamic content. For example, Akamai¹ and Fastly² have written blog posts about their moves towards caching requests with specific query strings and data content. This results in even more information being revealed to the CDN provider about the end user with each request made. The trend towards using CDNs for dynamic content indicates that the privacy risks that currently exist when using CDNs are likely to worsen in the future.

The threat of user profiling is compounded by the fact that many modern CDNs host a variety of different origin websites. An end user's IP address could act as an identifier to allow a CDN to track behavior across domains. Additionally, browser profiling, which is possible via many techniques [6], could make the tracking assessments more accurate, differentiating individual end users behind Network Address Translation (NAT). When CDNs have a large number of customers, as many do today, they can collect information about end users' browsing behavior across multiple contexts. In the wrong hands, this kind of large-scale user data collection can impact end users in a negative way. Price discrimination, where behavioral inferences are used to set prices, is an example of a negative effect that has already been observed in the wild on some e-commerce websites [7].

Privacy concerns of a similar nature to ours have already been raised with regard to Internet Service Providers (ISPs) [8]. These concerns are especially relevant given that recent broadband privacy rules were nullified in the United States [9]. However, little analysis has been done on CDN privacy threats to date. In this thesis, we aim to highlight and understand the nature of privacy threats from the use of CDNs. In this chapter, we perform measurements of CDN deployment on the top 250 Alexa-ranked websites in order to answer the following questions:

¹<https://blogs.akamai.com/2015/10/dynamic-page-caching-beyond-static-content.html>

²<https://www.fastly.com/blog/leveraging-your-cdn-cache-uncacheable-content>

- Do CDNs pose a significant privacy threat to Internet users?
- If so, which CDNs pose the biggest threats?
- How do CDNs specialize in the content they deliver, and how does specialization tend to exacerbate privacy threats?
- What are some ways to combat CDN privacy threats?

We describe techniques to determine which CDNs are deployed on a given webpage. We determine CDN usage in the top 250 websites, enabling quantification of the amount of information CDNs can collect across various contexts. In addition, we perform a categorical analysis showing that different CDNs tend to specialize in hosting content of particular types—this allows them a clearer picture of user behavior in specific domains. We discuss inferences that are possible based on this information, and consider the negative consequences for end users. Our findings show that CDNs are widely deployed, with Akamai hosting content on 36.8% of top websites and Fastly on 10.4%—this results in the ability of CDNs to aggregate a great amount of potentially sensitive user information across many websites.

We acknowledge that several challenges and open questions remain. Building tools to monitor and mitigate CDN privacy threats is a difficult problem, and will be the goal of future work in this area. We suggest possible avenues of future research concerning the design of these tools. Finally, we are releasing the scripts used for determining CDN deployment to the community³—we invite others to join us in the study of these privacy threats.

³<https://github.com/akashlevy/CDNDiscover>

1.2 Related Work

1.2.1 CDN measurement studies

The first measurement study of CDN usage was conducted by Krishnamurthy et al. in 2001 [1]. The study examined CDN techniques being employed, the extent to which CDNs were being used by popular origin server sites, the nature of content being offloaded by origin servers to CDNs, and relative CDN performance as compared to origin server performance. The authors used webcrawling techniques combined with `dig` (domain information groper) to make measurements, and found that common CDN techniques were URL rewriting and DNS redirection through canonical names (CNAMEs). They found that in the years 1999-2000, CDNs were already extremely prevalent for hosting static content on top websites—additionally, Akamai was overwhelmingly the most popular CDN then, hosting content on 165 out of the top 500 websites. The paper focused on a small set of CDNs that were popular during the time the measurement study was conducted. A similar study was conducted in 2008 [10]. Our study enhances the techniques described in previous work with the addition of Registration Data Access Protocol (RDAP) record lookups, as well as a more comprehensive list of hostname-CDN mappings. We additionally examine the tendency for CDNs to specialize in hosting certain categories of websites, and focus on the privacy threats associated with cross-site data aggregation.

1.2.2 Online privacy studies

In 2006, Krishnamurthy and Wills proposed the idea of a *privacy footprint*, allowing assessment and comparison of the diffusion of privacy information across a wide variety of sites [11]. In 2007 and 2009, Krishnamurthy et al. examined privacy protection mechanisms in place by browsers to reduce leakage of browsing information to third party websites [12, 13]. The Open Web Privacy Measurement framework (Open-

WPM) aims to rapidly detect, quantify, and characterize emerging online tracking behaviors [6]. Su et al. showed that it is possible to de-anonymize web browsing data with social networks [14].

1.2.3 Privacy-enhancing technologies

The Tor network is a privacy tool that directs Internet traffic through an encrypted overlay network consisting of more than seven thousand relays to conceal a user’s location and identity [5]. Tor provides anonymity to end users, preventing accurate/useful inferences about end users from being made in most contexts. The Tor network could present a viable solution to the CDN threats we examine in this thesis.

A line of research has proposed obfuscating user intent in online communications by introducing spurious cover traffic [15]. TrackMeNot was a preliminary implementation of this concept [16]. However, applying client-side obfuscation in the context of CDN services may introduce significant congestion on the Internet due to the added cover traffic.

Our work is novel in considering CDNs as a possible attack vector—to our knowledge, this is the first work that systematically investigates the privacy threats posed by CDNs from a measurement perspective. Our study paves the way for future work on monitoring and combating these threats.

1.3 Measuring the Privacy Threat

We note that servers often block or filter requests from automated web-scraping tools. To bypass these restrictions, we simulate an end user that connects to each of the root domains and requests web content. We use the Selenium framework [17] for this task—Selenium spawns a real Firefox browser that mimics an end user requesting a

resource. The browser allows us to maximize the likelihood of obtaining the same content an end user would see, and can execute additional tasks, such as interpreting scripts and requesting all the resources needed for a full rendering of a webpage.

Another advantage of a real browser such as Firefox, is the ability to install extensions that provide auxiliary functions—in our browser, we instrument Adblock Plus⁴ to eliminate all the links we do not wish to consider in our analysis. In particular, we ignore websites involved with advertising, analytics, APIs, and commonly-used libraries. To this end, we enhance Adblock’s built-in Easylist with an anti-tracking list⁵. We ignore these types of services to focus exclusively on content that CDNs serve in place of the origin server. For example, requesting a popular library like jQuery from a CDN would not be a privacy concern in our threat model, since this library could be included in multiple applications—its mere retrieval would not reveal any significant information about the end users’ behavior and interests.

Our client begins by making a request for the root domain webpage. Once the page is loaded, it parses the response and extracts all the `src` attributes, with the exception of `script` and `noscript` tags⁶. These attributes generally appear where references to static content (images, videos, CSS) are made. All the URLs found in this step are then processed in order to confirm if the static content they link is hosted on a CDN. We use a simple whitelist to determine which CDN is associated with a given hostname. Our whitelist is an adapted version of the one used by the WPO Foundation’s www.webpagetest.org [18] and is available as a JSON file in our Github repository⁷.

In addition to looking at `src` attributes, we also parse the links found in the HTML response using an XPath query for the `a` tags (corresponding to links). The `href` attributes of these tags are then added to a queue if they are in the same

⁴<https://adblockplus.org/>

⁵<https://github.com/eladkarako/hosts.eladkarako.com/>

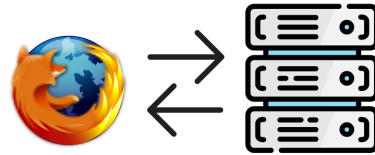
⁶These two tags typically refer to commonly-used libraries that do not pose privacy threats.

⁷The link to our project page is redacted for anonymization.

1. Retrieve Alexa Top Sites



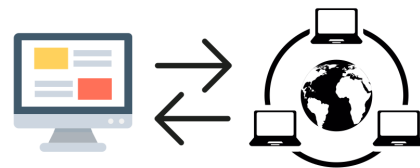
2. Use Firefox to navigate to root domain



3. Extract links

``

4. Perform CNAME/reverse DNS/
WHOIS lookups



5. Use whitelist to determine
CDN

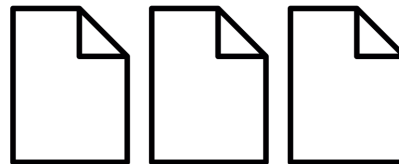


Figure 1.3: The steps involved in determining CDN usage in top websites.

domain as the root. We then perform a breadth-first search starting from the root page, repeating the process described above. We stop the search after 10 iterations, and move to the next root domain. We repeat this process for the top 250 Alexa domains. Figure 1.3 gives a visual representation of the steps involved in crawling, scraping and whitelisting.

1.3.1 CNAME Lookups

DNS redirection via CNAME rewriting is the most common way to make use of a CDN. A client will have a URL with a hostname that appears to belong to the origin server, but is instead redirected to a CDN-owned hostname. For example, a user visiting *www.mit.edu* is redirected to *www.mit.edu.edgekey.net* that belongs to Akamai.

This technique is typically easy to detect—a simple DNS query will generally return a CNAME record with the CDN’s hostname, which can be checked against our whitelist. Oftentimes, the origin websites will not have CNAME records, but their embedded URLs for static content will. This is what makes web crawling necessary for determining CDN usage in many cases. Occasionally, origin websites will directly rewrite the URLs in the HTML content they deliver. In scenarios where the origin website leverages URL rewriting for redirecting users to an edge server, it is possible to simply apply our whitelist to these URLs to identify the CDN operator.

1.3.2 Reverse DNS Lookups

Reverse DNS lookups are another technique for determining CDN usage. First, the hostname of the URL is resolved to an IP address, then a reverse DNS lookup is performed on this IP address—in some cases, this yields a hostname that is distinct from the CNAME, which may reveal the CDN being used. Similar to our CNAME technique, we use our whitelist to determine which CDN the PTR record belongs to.

1.3.3 RDAP Record Lookups

Registration Data Access Protocol (RDAP) record lookups are a technique for determining who owns a hostname or an IP address. For our purposes, looking up the hostname of a URL with RDAP will not provide much information since the host-

name typically belongs to the origin website. However, the IP address obtained by resolving the hostname could belong to a CDN—hence, an RDAP query for the IP address can provide information about the CDN operator.

We use the contact names listed in the RDAP record to determine which CDN is being used, matching the contact names with the CDNs listed in our whitelist. Note that this approach differs from our hostname matching in the other techniques utilized—here, we match literal CDN names (e.g. Akamai, Fastly, Cloudflare) as opposed to hostnames (e.g. edgesuite.net, fastly.net, cloudflare.net).

We choose RDAP as an alternative to WHOIS for a few reasons. Firstly, RDAP is faster and requests are usually not rate-limited as WHOIS queries are. Secondly, the response format is standardized, which makes parsing easy—for WHOIS records, it can be extremely difficult to make use of records because of the lack of standardization. Thirdly, we find that a large fraction of CDN vendors have RDAP entries for the IP addresses they own.

1.3.4 Determining CDN Usage with Confidence

Figure 1.3 provides an overview of our process for determining which websites host content on one or more CDNs. After downloading all the root pages from the top 250 websites in the Alexa list and extracting the tags from the HTML content, we leverage the three CDN resolution techniques mentioned earlier for generating a list of all the domains that are hosted on CDNs. If any of the Alexa-ranked root pages contains at least one link to a CDN-hosted domain in its HTML, we flag the top domain as hosting content on a CDN. As a next step, we use the list of the domains hosted on CDNs to determine which are the most popular providers.

For each top website, we consider only the CDNs we feel most confident the site is making use of. The confidence level is expressed in the *number* and *types* of links found while crawling each root page. If the origin domain is hosted on a CDN, we are

highly confident that the website is using that CDN. Elsewhere, we require *four* cross-domain links to CDN-hosted content to ensure that the website is really offloading a significant portion of its origin content to that CDN—oftentimes, one or two foreign resources on a webpage will be hosted on a CDN that the domain is not actually a client of. For example, the `www.bookmyshow.com` homepage has two embedded links to external content on Akamai, but these resources are anomalies—Cloudflare hosts the majority of the content. Requiring four links was found to significantly reduce the likelihood of falsely identifying CDN usage in cases like this.

Table 1.1: Alexa-ranked English websites and their CDNs

#	Domain	1	2	#	Domain	1	2	#	Domain	1	2
1	google.com			41	bbc.co.uk			81	ncbi.nlm.n...		
2	youtube.com			42	cnn.com			82	4chan.org		
3	facebook.com			43	soundcloud...			83	walmart.com		
4	mail.googl...			44	amazon.co.uk			84	store.stea...		
5	yahoo.com			45	booking.com			85	support.mi...		
6	reddit.com			46	mozilla.org			86	huffington...		
7	google.co.in			47	ask.com			87	indiatimes...		
8	docs.googl...			48	xnxx.com			88	bbc.com		
9	en.wikiped...			49	espn.com			89	bankofamer...		
10	twitter.com			50	nytimes.com			90	myway.com		
11	amazon.com			51	vimeo.com			91	mega.nz		
12	search.yah...			52	blogger.com			92	trello.com		
13	google.co.uk			53	ebay.co.uk			93	news.yahoo...		
14	linkedin.com			54	bet365.com			94	wellsfargo...		
15	mail.yahoo...			55	salesforce...			95	zillow.com		
16	netflix.com			56	spotify.com			96	weather.com		
17	imgur.com			57	chaturbate...			97	news.googl...		
18	translate....			58	theguardia...			98	cricbuzz.com		
19	ebay.com			59	slideshare...			99	tripadviso...		
20	bing.com			60	chase.com			100	youporn.com		
21	wordpress.com			61	dailymail....			101	livejourna...		
22	msn.com			62	answers.ya...			102	thesaurus.com		
23	twitch.tv			63	mediafire.com			103	wordrefere...		
24	tumblr.com			64	cnet.com			104	archive.org		
25	microsoft.com			65	deviantart...			105	irctc.co.in		
26	xvideos.com			66	indeed.com			106	weebly.com		
27	stackoverflow...			67	livejasmin...			107	forbes.com		
28	imdb.com			68	flipkart.com			108	ikea.com		
29	office.com			69	9gag.com			109	google.ie		
30	pinterest.com			70	nih.gov			110	foxnews.com		
31	amazon.co.jp			71	wikihow.com			111	intuit.com		
32	github.com			72	etsy.com			112	google.com		
33	wikia.com			73	bbc.co.uk			113	speedtest.net		
34	apple.com			74	godaddy.com			114	msdn.micro...		
35	google.com.au			75	nlm.nih.gov			115	feedly.com		
36	paypal.com			76	battle.net			116	aol.com		
37	adobe.com			77	alibaba.com			117	blackboard...		
38	play.googl...			78	roblox.com			118	ign.com		
39	dropbox.com			79	washington...			119	businessin...		
40	plus.googl...			80	yelp.com			120	shuttersto...		

#	Domain	1	2	#	Domain	1	2	#	Domain	1	2
121	skype.com			165	timesofind...			209	4shared.com		
122	researchga...			166	mit.edu			210	pixabay.com		
123	sourceforg...			167	reverso.net			211	gsmarena.com		
124	scribd.com			168	target.com			212	java.com		
125	flickr.com			169	americanex...			213	nike.com		
126	rt.com			170	gizmodo.com			214	cambridge.org		
127	asos.com			171	vice.com			215	wsj.com		
128	espnricin...			172	bloomberg.com			216	google.rs		
129	bestbuy.com			173	aws.amazon...			217	wowhead.com		
130	goodreads.com			174	duckduckgo...			218	zoho.com		
131	gamefaqs.com			175	box.com			219	onlinelibr...		
132	bongacams.com			176	reuters.com			220	khanacadem...		
133	oracle.com			177	ups.com			221	google.com		
134	samsung.com			178	fiverr.com			222	bleacherre...		
135	download.c...			179	ultimate-g...			223	groupon.com		
136	xfinity.com			180	azlyrics.com			224	naukri.com		
137	leagueofle...			181	livescore.com			225	avg.com		
138	wordpress.org			182	dell.com			226	springer.com		
139	mailchimp.com			183	prezi.com			227	porn.com		
140	telegraph....			184	bookmyshow...			228	discogs.com		
141	hp.com			185	siteadviso...			229	surveymonk...		
142	accuweathe...			186	rottentoma...			230	in.yahoo.com		
143	pandora.com			187	flirt4free...			231	investoped...		
144	finance.ya...			188	webmd.com			232	mail.aol.com		
145	support.mo...			189	udemy.com			233	kijiji.ca		
146	independen...			190	alexa.com			234	engadget.com		
147	amazon.ca			191	google.co.nz			235	uploaded.net		
148	homedepot.com			192	icicibank.com			236	hm.com		
149	evernote.com			193	rediff.com			237	badoo.com		
150	ndtv.com			194	patreon.com			238	ibm.com		
151	utorrent.com			195	goal.com			239	google.com		
152	shopify.com			196	billdesk.com			240	nhl.com		
153	hulu.com			197	theverge.com			241	npr.org		
154	zendesk.com			198	wiley.com			242	irs.gov		
155	usps.com			199	wiktionary...			243	money.cnn.com		
156	nba.com			200	xda-develo...			244	att.com		
157	mlb.com			201	sports.yah...			245	asus.com		
158	hdfcbank.com			202	tweetdeck...			246	humblebund...		
159	go.com			203	ebay.com.au			247	groups.goo...		
160	edition.cn...			204	playstatio...			248	expedia.com		
161	capitalone...			205	filehippo.com			249	squarespac...		
162	urbandicti...			206	fedex.com			250	ebay.in		
163	behance.net			207	thefreedic...						
164	sciencedir...			208	usatoday.com						

CDNs:


































 Akamai	 Alibaba	 Amazon CloudFront
 CDNetworks	 Cloudflare	 Edgecast
 Fastly	 Highwinds	 Incapsula
 Instart Logic	 Internap	 LeaseWeb CDN
 Level 3	 Limelight	 MaxCDN
 Microsoft Azure	 Rackspace	 Reflected Networks

Table 1.2: Aggregate Statistics for CDN Usage

#	CDN	No. Domains	% CDN	% total
1	 Akamai	92	62.2%	36.8%
2	 Fastly	26	17.6%	10.4%
3	 Amazon CloudFront	9	6.1%	3.6%
4	 Cloudflare	4	2.7%	1.6%
5	 Instart Logic	3	2.0%	1.2%
6	 Limelight	2	1.4%	0.8%
7	 Level 3	2	1.4%	0.8%
8	 Highwinds	2	1.4%	0.8%
9	 MaxCDN	2	1.4%	0.8%
10	 Edgecast	1	0.7%	0.4%
11	 CDNetworks	1	0.7%	0.4%
12	 Incapsula	1	0.7%	0.4%
13	 LeaseWeb CDN	1	0.7%	0.4%
14	 Alibaba	1	0.7%	0.4%
15	 Microsoft Azure	1	0.7%	0.4%

Once the links are finished being resolved to their respective CDNs, we can compile a list of which websites use CDNs to determine which are the most popular providers. For each top website, we consider only the CDNs we feel most confident the site is making use of. Our confidence is based on the number and types of links found while crawling each page. A confidence score is kept for each CDN. If a link is a local link (i.e. belonging to the root domain), 5 is added to the confidence score of the CDN corresponding to that link. If a link is a non-local link, 1 is added to the confidence score. We threshold our confidence at 4 for reporting it in our results. The reason we use this scoring system is because local links hosted on a CDN are more likely to be accurate, while non-local domains are often APIs, libraries, or other 3rd-party services that slip through our Adblock filtering.

Table 1.1 is a list of the top 250 Alexa-ranked websites and their top two CDNs. Table 1.2 provides aggregate statistics about the most frequently used CDNs. The first column indicates the popularity ranking of the CDN and the second column is the name of the CDN. The third column is the total number of top websites (out of 250) that used that CDN and the fourth column is the percentage of CDN-based websites using that CDN. The fifth column is the percentage of all websites using that CDN.



1.4 Validating Measurements

After making measurements, we attempted to validate some of them using client lists for Akamai and Fastly. These lists provide ground truth for evaluating our study. Our list for Akamai consisted of 169 websites listed on its customer list webpage⁸. Our list for Fastly consisted of 64 websites displayed on its customers webpage⁹. There is no guarantee that the customer listings on these websites are up-to-date—

⁸<https://www.akamai.com/us/en/customer-case-studies/our-customers-list.jsp>

⁹<https://www.fastly.com/customers>

Table 1.3: CDN Measurement Validation Results

CDN	Correct	Missing	Incorrect	Total
 Akamai	111 (65.68%)	46 (27.22%)	12 (7.10%)	169
 Fastly	46 (71.88%)	9 (14.06%)	9 (14.06%)	64

companies will often switch CDN providers, and it is possible that the entries are not updated when this happens. For example, Firebase recently moved to a Google domain (<https://firebase.google.com/>) yet it was still listed on Fastly’s customer list. Nevertheless, we tested our client on the domains we collected for an estimate of what kind of accuracy/coverage we can expect in our results, keeping in mind that these estimates are a lower bound.

Our results are presented in Table 1.3. We have fairly high accuracy for both Akamai and Fastly (65.58% and 71.88% respectively), though our coverage in Akamai is lower (27.22% missing vs. 14.06% for Fastly). Our Akamai measurements showed a lower false positive rate than Fastly (see *Incorrect* column). While our validation measurements do not indicate full accuracy, they show that our techniques work in most cases and also indicate quite low false positive rates (7.10% for Akamai, 14.06% for Fastly).

1.5 Categorical Analysis

In order to determine how CDN usage varies across different classes of websites, we also performed a categorical analysis using the techniques described above. We use the top 25 English websites in 16 top-level categories provided by Alexa. The categories are: Adult, Arts, Business, Computers, Games, Health, Home, Kids and Teens, News, Recreation, Reference, Regional, Science, Shopping, Society, and Sports. For each category, we perform a web crawl of the top 25 domains and report CDN usage. Figure 1.4 provides a radar chart which illustrates the results.

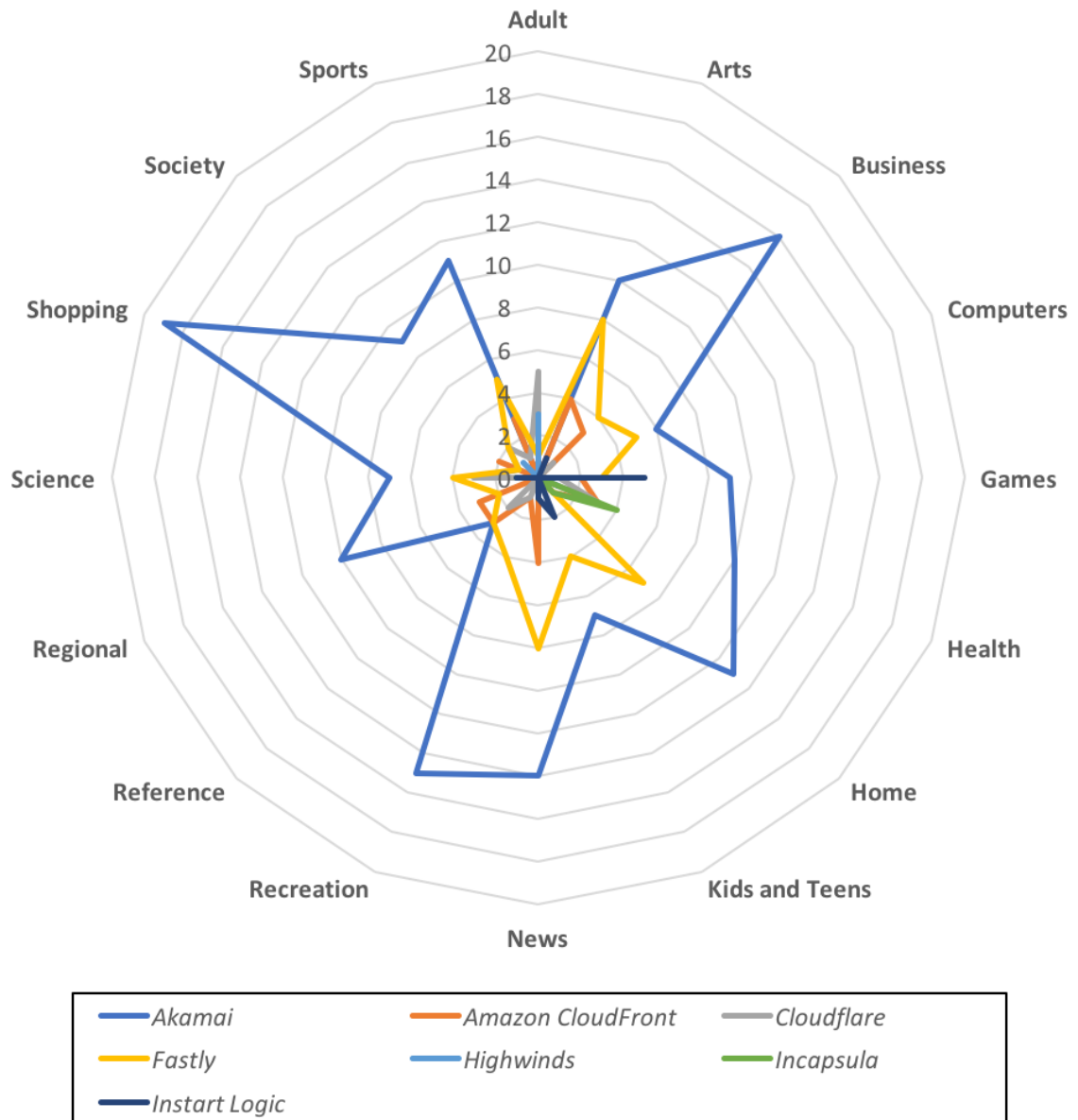


Figure 1.4: A radar chart of CDN usage in the top 25 English websites of each top-level category provided by Alexa. Each CDN appears to have classes of websites which they specialize in hosting.

We see that Akamai specializes in shopping and business, and Fastly specializes in news, home, and arts websites—neither host any adult content. Instart Logic hosts five of the top 25 gaming sites, while Incapsula specializes in adult and health content. These categorical specializations have privacy implications for end users, since browsing data can be collected on users across multiple contexts. Utilizing distinct observations made over several sites could enhance the quality, and hence value, of the inferences made about these users.

1.6 Analysis of CDN Measurement

We find that CDNs are quite prevalent in top websites today that garner a large amount of daily traffic—out of the 250 top English Alexa websites, 164 (65.6%) use a third-party CDN. Akamai is shown to be the clear leader, and on its own, serves content on over a third of top sites. Fastly, CloudFront, and Edgecast also deliver a significant amount of top website content.

We observe that many top websites use multiple CDNs. This technique may further increase the performance of the website by enabling selection from a larger set of servers. For example, Business Insider hosts half of its static content on Akamai and half on Fastly:

```
static3.businessinsider.com:  f.global-ssl.fastly.net.  
static5.businessinsider.com:  f.global-ssl.fastly.net.  
static1.businessinsider.com:  f.global-ssl.fastly.net.  
static4.businessinsider.com:  ustbistatic.edgekey.net.  
static6.businessinsider.com:  ustbistatic.edgekey.net.  
static2.businessinsider.com:  ustbistatic.edgekey.net.
```

By looking at the types of websites hosted, we see that Akamai is the preferred option for large corporations such as Microsoft, Apple, eBay, and Paypal. On the other hand, Fastly specializes in media sites, delivering content on behalf of Reddit, CNN, NYTimes, Vice, The Guardian, BuzzFeed, India Times, and Daily Mail.

It is interesting to note that Akamai hosts content for many companies that are involved in the transfer of money. Direct access to content of services such as Paypal and eBay could allow the provider to infer sensitive information about end users' finances. Sharing this knowledge with third parties could have negative consequences for users if this information is used for illicit practices (e.g., price discrimination).

Fastly's focus on media sites is another potential threat for end users. Correlation amongst services and news sites that users most frequently access could be leverage for learning political preferences and topics that users are interested in.

1.7 Conclusion

In this section, we discuss how CDNs present an attack vector for compromising the privacy of Internet users. Our measurement of CDN adoption across the top 250 Alexa-ranked websites shows how these networks are widely deployed among the most popular services. We argue that since CDN providers have access to the content they deliver and the IP address of end users, they are in a strategic position to classify these users' interests, even when communications are encrypted.

In our exploratory study, we observe that different providers tend to specialize in hosting content on websites that provide similar services. Specifically in the case of Akamai and Fastly, this includes applications used for money transfer and news services. The data collected can be abused to infer information about users across multiple sources—the combination of observations made across several sites can increase sensitivity, scope, and accuracy of behavioral inferences.

The widespread adoption of CDNs among top websites represents an important threat that has yet to be studied in depth. Our study elucidates some pressing issues related to CDNs and privacy, and we hope that it spurs further research and awareness in this area.

Chapter 2

Large-Scale Measurement of CDN Deployment

2.1 Introduction

The Internet has become increasingly interdependent in modern times—there has been a trend towards greater deployment of advertising/tracking technologies, and an increased reliance on third-party platforms. This pattern has been fueled by the economics and performance demands of the web today. Quite remarkably, a recent study showed that the median number of external resources loaded per webpage has doubled in the last five years [19]. Understanding the complex dependency graph of the Internet has therefore become very important to consider in tackling recent challenges related to web security/privacy.

Prior work has examined the implications of the tangled web. Measurement of HTTPS key sharing between origin hosts and CDN providers has shown that it is a common practice on the Internet today, giving hosting providers the ability to decrypt traffic of origin hosts—inevitably, this makes them a prime target for attacks aiming to compromise private keys [3]. Besides key sharing, widespread deployment

of shared content across multiple websites leads to an increased attack surface—in 2013, for example, the BootstrapCDN was compromised, and a malicious payload was injected into a script that was deployed on thousands of websites¹. This incident and others similar to it have motivated the need for the development of Subresource Integrity (SRI) to verify content delivered from third-parties [20].

Many papers have examined privacy leakage through several techniques that are commonly used by advertising platforms and analytics frameworks. They have explored the use of cookies and supercookies, abuse of web APIs, and leakage of personal identifiers to third-parties [6] [21] [22]. In this paper, we explore passive privacy attacks that are possible without the use of cookies on the top-million Alexa websites². We show that it is possible for organizations to leverage widely-deployed content to obtain a global picture of web browsing, obtaining individual users’ browsing history with high accuracy.

Privacy concerns of a similar nature to ours have already been raised with regard to Internet Service Providers (ISPs) [8]. These concerns are especially relevant given that recent broadband privacy rules were nullified in the United States [9]. However, less analysis has been done on privacy threats that persist without the use of cookies. In this section, we provide a preliminary analysis of threats to end-user privacy caused by third-party offload, and make recommendations/suggest future directions of research based on our measurements.

¹<https://www.maxcdn.com/blog/bootstrapcdn-security-post-mortem/>

²Alexa Top 1M Sites: <http://s3.amazonaws.com/alexa-static/top-1m.csv.zip>

2.2 Related Work

2.2.1 Web privacy measurement tools

A study by Finamore et al. showed that around 75% of Internet traffic from a representative sample of users was being served by the top 15 organizations. [23]. A tool called *FourthParty* was developed to characterize web tracking using a headless browser [24]. *FPDetective* is another tool that instruments the Selenium framework to study tracking and fingerprinting trends. This paper provides a comprehensive overview of browser fingerprinting techniques utilized in the wild. The more recent *Open Web Privacy Measurement (OpenWPM)* framework aims to rapidly detect, quantify, and characterize emerging online tracking behaviors in a browser setting approximating that of a real user [6]. This paper built upon FPDetective’s pioneering work with the addition of stateful measurements to examine cookie syncing and other related phenomena.

2.2.2 Hostname/IP to organization mapping

There has been a strong desire in the research community for a robust way to map hostnames and IP addresses to the organizations responsible for them. WHOIS records are intended to provide human-readable information towards this end—however, in practice, WHOIS records can be inconsistent and often describe organizations at varying levels of granularity. Moreover, the records’ format is not standardized, making them difficult to parse for machines—much recent research has been directed towards accurate parsing techniques at scale. For example, Liu et al. developed a statistical model for parsing WHOIS records that learns from labeled examples using a conditional random field (CRF) with a small number of hidden states, a large number of domain-specific features, and parameters that are estimated by efficient dynamic-programming procedures for probabilistic infer-

ence [25]—ultimately, they claimed they were able to achieve well over 99% accuracy with their technique.

The Internet Engineering Task Force (IETF) has also made an attempt to address the shortcomings of WHOIS records with the introduction of *Registration Data Access Protocol (RDAP)*. This protocol leverages the wide-used JSON format to standardize the responses to requests for registration data records [26]. This protocol is in its infancy but is picking up use steadily.

Other attempts at hostname/IP to organization mapping have relied on whitelists, such as was done in Krishnamurthy et al.’s original measurement study on CDN deployment in 2001 [1]. A more comprehensive up-to-date list for mapping hostnames and response headers to CDN’s is maintained by WebPageTest at webpagetest.org [27]. Researchers have also utilized information about autonomous systems (ASes) to use organizations—Cai et al. describe some preliminary techniques for identifying multi-AS organizations on the Internet [28]. The study by Cangliosi et al. on private key sharing in the HTTPS ecosystem utilized AS and WHOIS data and performed clustering techniques on it to understand key sharing across organizations [29].

2.3 OpenWPM: Automated Large-Scale Privacy Measurement

In this study, we made use of data collected on the top million websites made available through OpenWPM, which uses an automated version of a complete consumer browser. The framework runs in the cloud (usually on Amazon EC2 instances) and operates in a highly parallel fashion, with mechanisms for automatic failure recovery, and tracking of granular information in the browser. Below we provide a summary of how OpenWPM works and why it is useful for our study.

OpenWPM focuses on three main components of web measurement: simulating users, recording data collected from websites, and analysis. This platform fully automates user simulation and data collection and makes the task of analysis simpler for researchers. It is open source and is available in its entirety on GitHub³.

It also supports stateful measurements, although we do not utilize them in this study. This platform is unique—while native codebase changes on other platforms require constant merges as the upstream codebase evolves and complete rewrites to support alternative browsers, OpenWPM is designed to avoid these challenges. Browser automation and data collection are divided into browser managers (which automate individual browsers), a task manager which distributes commands to the browser managers which may be on multiple machines, and a data aggregator. A figure describing OpenWPM’s operation is given in Figure 2.1. The entire platform is built using Python and Python libraries.

2.3.1 Browser driver

OpenWPM uses Selenium, a web driver for Firefox, Chrome, and several other browsers. Firefox is used for the crawl we analyze in this study. Since Firefox is a mainstream browser, using it for measurement results in a very high probability that the data will correspond to what a real-world user would see.

2.3.2 Browser managers

These provide stability for automated measurement. Many unpredictable events can halt progress or cause data loss/corruption during a web crawl. For example, Selenium frequently halts indefinitely due to its blocking API. With browser managers, such problems can be abstracted away. Each browser manager works with a Selenium instance in a specified configuration and converts high-level commands into

³<https://github.com/citp/OpenWPM>

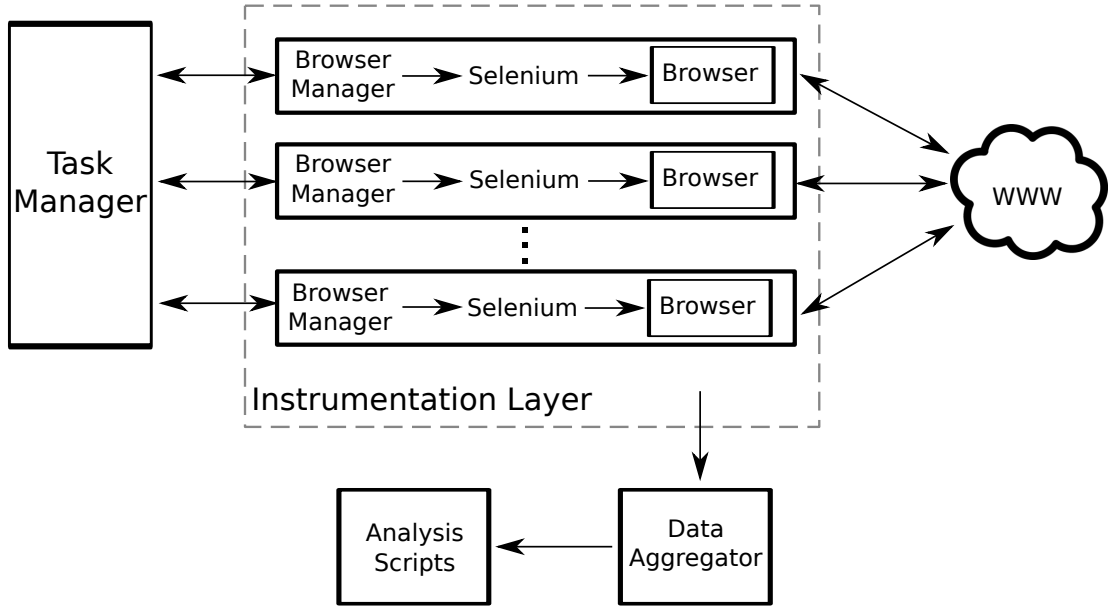


Figure 2.1: An overview of how OpenWPM works (adapted from [6]). The task manager monitors browser managers, which convert high-level commands into automated browser actions. The data aggregator receives and pre-processes data from instrumentation.

specific lower-level commands to be executed by Selenium—this enables recovery from browser failures. Running each browser manager as a separate process enables isolation of browser failures.

2.3.3 Task manager

This component is another abstraction that enables scalability. It provides an interface for distributing tasks across multiple browsers simultaneously. Each command is launched in a per-browser command execution thread. The command-execution thread handles errors in its corresponding browser manager automatically. If the browser manager fails in any way, the thread enters a crash recovery routine. Here, the manager backs up the current browser profile, kills the currently running processes, and loads the (stateful) backup into a fresh browser with the same configuration.

2.3.4 Data aggregator

This component makes provisions for repeatability. Repeatability is accomplished by logging data in a systematic fashion. All data is aggregated centrally—the aggregator has its own process, and is accessed through a socket interface.

2.3.5 Instrumentation

This component supports global and reproducible measurement. In particular, OpenWPM provides:

1. Raw data on disk
2. Raw data at the network level with an HTTP proxy with MITMProxy
3. Raw data at the JavaScript level with a Firefox extension

Overall, these features enable the framework to guarantee high coverage of a browser’s interaction with the web and the system.

2.4 Measuring Third-party Content Deployment

We evaluate third-party content deployment on the top million Alexa websites using five different techniques on the results of the 2016 OpenWPM web crawl:

1. Hostname whitelisting
2. CNAME whitelisting
3. Reverse DNS hostname whitelisting
4. Response header whitelisting
5. ASN to organization mapping

Table 2.1: Hostname Whitelisting: Organization Counts

Third-party	Count
Google	632534
Facebook	317429
Amazon CloudFront	73905
Reapleaf	60060
Twitter	51900
WordPress	51734
Cloudflare	48045
Akamai	12041
Yahoo	11348
KeyCDN	8754
NetDNA	6145
jsDelivr	5646
Microsoft Azure	2600
Taobao	1552
CDN77	790
Fastly	727
Edgecast	524
Limelight	506
Cachefly	489
Highwinds	289
Internap	210
cubeCDN	191
OnApp	160
CDNetworks	136
Rackspace	120
Yottaa	112
Advanced Hosters CDN	68
Level 3	67
Medianova	65
CDNvideo	64
Instart Logic	56
section.io	50
LeaseWeb CDN	38
NGENIX	34
Incapsula	32
HiberniaCDN	27
BitGravity	24
ReSRC.it	22
CDNsun	16
Netlify	11
BO.LT	2
Rev Software	2

We compare the coverage of these different techniques and cross-validate our measurements to determine robustness. We then analyze our findings in the context of end user privacy. Tables 2.1-2.5 show the organizations and the number of domains hosted on each as determined by the five methods listed above.

We can see that the hostname methods typically find Google, Facebook, Amazon, and Akamai in the top few third-parties. Request header whitelisting appears to have very low coverage (confirmed in the next section). The AS-based approach finds

Table 2.2: CNAME Whitelisting: Organization Counts

Third-party	Count
Google	768076
Facebook	312729
Akamai	250482
Fastly	136666
NetDNA	135360
Amazon CloudFront	94554
Realeaf	59937
Edgecast	59771
Twitter	41559
Cedexis	25772
KeyCDN	13999
Highwinds	10868
Incapsula	7721
CDN77	6743
Microsoft Azure	6656
StackPath	5138
Level 3	5129
Yahoo	5120
CDNetworks	4637
Limelight	4219
CDNvideo	2804
Cachefly	2623
ChinaNetCenter	2533
Advanced Hosters CDN	1318
Taobao	1077
Instart Logic	992
Internap	964
OnApp	559
Reflected Networks	535
CDNsun	534
Rackspace	527
LeaseWeb CDN	335
ChinaCache	299
NGENIX	298
cubeCDN	293
Azion	264
BitGravity	206
Medianova	190
Yottaa	150
SwiftCDN	122
Zenedge	94
Netlify	82
Rev Software	82
HiberniaCDN	56
section.io	55
GoCache	47
Aryaka	35
UnicornCDN	33
KINX CDN	20
BunnyCDN	9
Optimal CDN	7
Mirror Image	4
SFR	4
NYI FTW	3
Hosting4CDN	3
Cloudflare	1
Bison Grid	1
BO.LT	1

Table 2.3: Reverse DNS Whitelisting: Organization Counts

Third-party	Count
Facebook	318197
Akamai	284541
Amazon CloudFront	160447
Google	89657
WordPress	19226
Highwinds	16411
Incapsula	10912
Cachefly	2459
Instart Logic	484
NGENIX	306
BitGravity	224
Yahoo	149
HiberniaCDN	82
cubeCDN	44
Cloudflare	32
Edgecast	2

Table 2.4: Request Header Whitelisting: Organization Counts

Third-party	Count
Twitter	112873
Highwinds	38388
CDNetworks	29353
Incapsula	14218
OVH CDN	2538
Rev Software	78
Aryaka	30
section.io	27
Google	16
Amazon CloudFront	1
Caspowa	1

Table 2.5: ASN Lookups: Organization Counts

AS Organization	Count
Amazon.com, Inc.	579217
CloudFlare, Inc.	280558
Google Inc.	102579
OVH SAS	69947
Hetzner Online GmbH	55969
GoDaddy.com, LLC	46763
Fastly	29740
Alibaba (China) Technology Co., Ltd.	28141
Adobe Systems Inc.	26080
Unified Layer	25775
Rackspace Hosting	24603
Automattic, Inc	24030
Akamai International B.V.	23472
Linode, LLC	23112
Digital Ocean, Inc.	22866
SoftLayer Technologies Inc.	22616
SAKURA Internet Inc.	21896
Aruba S.p.A.	21251
Liquid Web, L.L.C	20035
ServerStack, Inc.	20025
Microsoft Corporation	18261
Host Europe GmbH	17350
CyrusOne LLC	17012
Akamai Technologies, Inc.	16890
1&1 Internet SE	16467
No.31,Jin-rong Street	15726
CNCGROUP Jitong IP network	15705
Yahoo!	13894
Incapsula Inc	13236
Amazon.com Tech Telecom	12793
Confluence Networks Inc	12525
VKontakte Ltd	12036
Squarespace, Inc.	11827
SingleHop, Inc.	11793
Limited liability company Mail.Ru	11123
Shopify, Inc.	10187
Level 3 Communications, Inc.	10071

Cloudflare-hosted websites, which are typically overlooked by the other methods. This indicates that we may want to cross-validate our methods.

We also see that there may be many privacy implications associated with having such a high frequency of resource loading from each third party. Google resources are detected on 768,076 of the top million websites, which means that they receive request traffic from over three-quarters of the top million domains. Amazon appears to receive 579,217 requests according to the AS-based measurement approach, making them involved in loading resources for over half of the top million websites. These trends are concerning as it is possible that these organizations can learn about users’ interests based on the content they request. We elaborate further on this idea in the following chapters.

2.5 Cross-Validation of Measurement Techniques

We also wish to determine how often the different measurement techniques agree and differ. To this end, we compared the CDN lists produced by different techniques—we present the results in Figure 2.2 as a Venn diagram giving the average number of links per top-million domain on which the methods agreed/disagreed. For simplicity, we combined three of the methods (hostname, CNAME, reverse DNS whitelisting) into a single “Hostname” method—we also provide a cross-validation of these three methods individually in Figure 2.3.

2.5.1 Comparison of AS/hostname/header-based techniques

We found that on average, 0.334 CDNs were found per domain by hostname whitelisting methods that were not found by either the AS-organization map or the header whitelisting method. We found 1.642 CDNs on average by the AS-based method that were not found by either the hostname or the header method. The header method

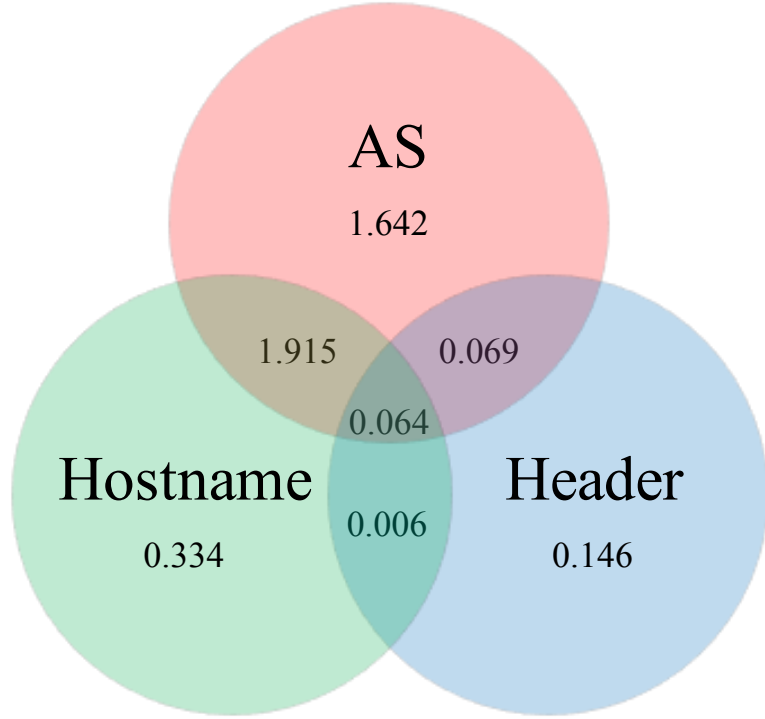


Figure 2.2: Cross-validation of different CDN measurement techniques. “AS” represents the technique for AS-organization mapping, “Hostname” represents the hostname/CNAME/reverse DNS whitelisting methods (combined), and “Header” represents the header whitelisting method. The numbers assigned to the different intersections represent the mean number of CDNs the corresponding techniques collectively found. A comparison between hostname/CNAME/reverse DNS whitelisting methods is given in Figure 2.3.

discovered 0.146 CDNs on average that were not picked up by either the AS or hostname methods. There were very few cases in which the header and the hostname methods agreed on a CDN but the AS method disagreed (0.006 CDNs/domain on average). The header and AS methods tended to agree more with 0.069 CDNs/domain picked up by both AS and header methods but not the hostname methods, and 0.064 CDNs/domain picked up by all three. Lastly, the AS and hostname methods showed strong agreement, with 1.915 CDNs picked up per domain (that were not picked up by the header method).

The results show that the header method provides low coverage, while the AS and hostname methods tend to agree on many of the CDNs. We believe that the

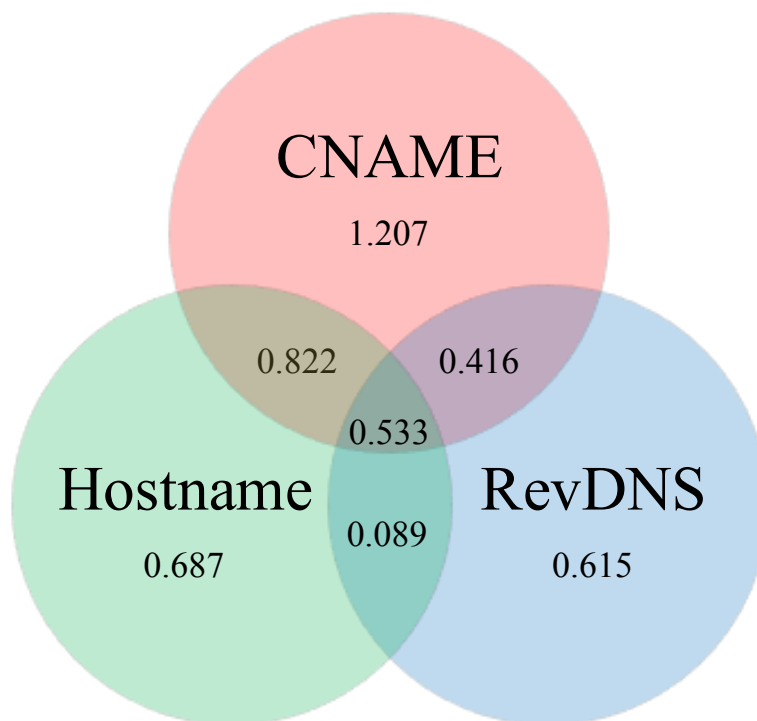


Figure 2.3: Cross-validation of different hostname whitelisting measurement techniques. The numbers assigned to the different intersections represent the mean number of CDNs the corresponding techniques collectively found. The results indicate that each method has coverage of a largely independent set of CDNs.

AS-based approach may often pick up organizations that are not CDNs. Overall, a good heuristic that provides good confidence might be to determine that a CDN is in usage when at least two of the three methods agree that a CDN is in use. By this heuristic, each of the top million Alexa domains use approximately two CDNs on their root pages.

2.5.2 Comparison of hostname-based techniques

Our second cross-validation figure (Figure 2.3) shows us that the different hostname whitelisting techniques produce somewhat independent results. These methods are (1) direct hostname whitelisting, (2) CNAME whitelisting, and (3) reverse DNS hostname whitelisting, and they are described in more detail in Chapter 1.

2.6 Conclusion

In this chapter, we have analyzed five different techniques for measuring CDN usage on the Internet. We implement these five techniques on a fully-instrumented dynamic web crawl using OpenWPM. We then show that the aggregation of data in the top third-party organizations poses concerns for end-user privacy. We perform a cross-analysis of the different methods for measuring CDN usage and find that a combination of methods is probably best for making tradeoffs between accuracy and coverage in determining which CDNs are in use.

Chapter 3

Privacy Threats from Weak Referrer Policy

3.1 Introduction

The threat of user profiling on the Internet is compounded by the fact that many third-parties provide content to a variety of different origin websites. Third-parties are capable of determining the origin webpage by exploiting the **Referer** headers that tend to appear in requests to CDNs based on the *Referrer Policy* [30]. Even when policies are put in place to prevent **Referer** headers from being sent, **XmlHttpRequests** (XHRs) can generate **Origin** headers to enable Cross-Origin Resource Sharing (CORS)—this can leak a limited amount of information about which domain an end user is visiting. Browsing history is a sensitive but valuable resource for advertisers—in the past, trackers/advertisers have exploited vulnerabilities in the way hyperlinks are rendered in browsers in order to hijack users’ browsing history [31]. Passive fingerprinting techniques based on IP address, operating system, user agent, language, and HTTP accept headers, as well as more sophisticated traffic fingerprinting techniques, could make tracking assessments more accurate, differentiating individual users behind Net-

work Address Translation (NAT). In the wrong hands, this kind of large-scale user data collection can impact end users in an acutely negative way. Price discrimination, where behavioral inferences are used to set prices, is an example of a negative effect that has already been observed in the wild on some e-commerce websites [7].

3.2 Related Work

3.2.1 Studies on third-party content and privacy

Kumar et al. published a study on security challenges in the modern web due to its increasingly complex dependency graph [19]. The study had many interesting findings: (1) just over 90% of the top million sites have external dependencies, and more than two thirds of all resources are loaded from external sites, (2) at least 20% of sites depend on content loaded from Google, Facebook, Amazon, Cloudflare, and Akamai ASes, and (3) websites load a median 23 external resources from 9 external domains, 3 external ASes, and 1 foreign country. They also note that third-party content deployment presents a barrier to full HTTPS deployment.

In 2006, Krishnamurthy and Wills proposed the idea of a *privacy footprint*, allowing assessment and comparison of the diffusion of privacy information across a wide variety of sites [11]. In 2007 and 2009, two papers examined privacy protection mechanisms in place by browsers to reduce leakage of browsing information to third party websites [12] [13]. In 2009, another study by Krishnamurthy et al. evaluated the leakage of personally identifiable information (PII) via social networks [32]. In 2011, these authors examined over 100 popular sites in several categories to see if these sites leak private information to prominent aggregators, and showed leakage in sites for every category examined; 56% of the sites directly leak pieces of private information (75% if leakage of a site user ID is included) [33]. The authors criticize an FTC report that they argue fails to emphasize the significance of third-party user

information leakage through negligence on the part of first-party websites, and recommend that (1) first-party websites never pass sensitive information through URLs and (2) these websites take the initiative to configure their referrer policies to protect end user privacy.

3.2.2 Studies on traffic fingerprinting and defenses against it

There have been many privacy attacks demonstrated on web traffic using fingerprinting techniques in previous work. Some prominent examples include Panchenko et al.’s attack on the Tor network [34], k -fingerprinting [35] which leverages a random forest algorithm to classify web traffic, and Wang and Goldberg’s SVM technique for fingerprinting Tor web traffic [36]. These techniques can be extended to incorporate the additional information available to third-parties about their clients.

There have also been proposals on potential defense techniques. Some researchers have proposed injecting random traffic to prevent accurate/useful inferences from being made [15]. TrackMeNot was a preliminary implementation of this concept [16]. However, there have been many criticisms of TrackMeNot (TMN) [37] [38]. The critics claim that the privacy guarantees of TMN can be broken with basic machine learning classifiers, hence rendering this implementation of a privacy tool ineffective.

Two 2012 papers and a 2015 paper showed that Netflix and Hulu, two popular video streaming services, have their movies and TV shows hosted on multiple CDNs [39] [40] [41]. Hosting content on multiple CDNs is a strategy for further increasing performance for end users. As usual, this is done by choosing a server that is likely to be the fastest (usually based on geographical location)—however, the set of servers to choose from is larger. It is possible that using multiple CDNs could reduce the privacy threats associated with large-scale data aggregation within a single CDN.

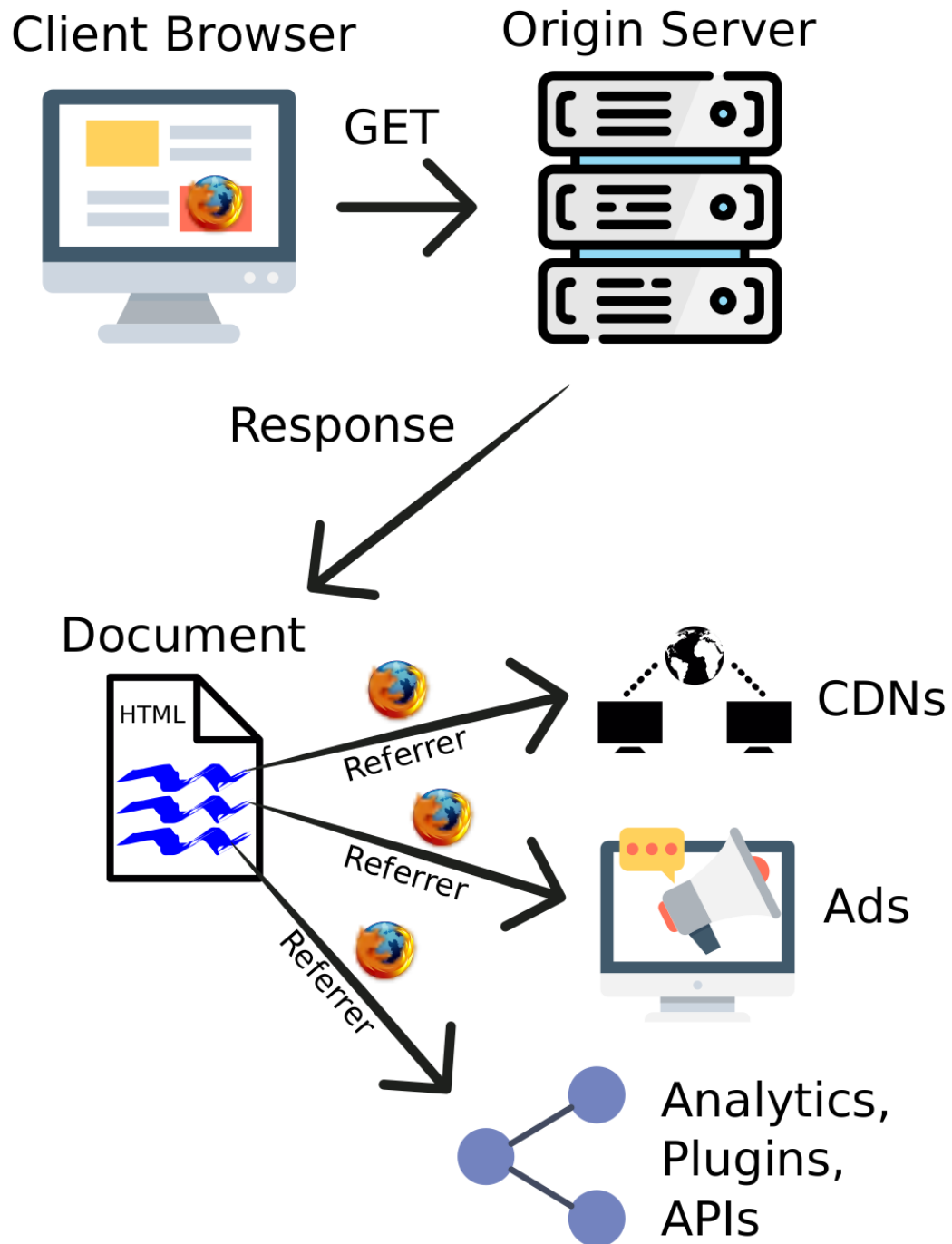


Figure 3.1: A depiction of how referrer information is propagated to third-parties. The browser loads dependencies of a request page and sends a `Referer` header that leaks information about the document being requested.

3.3 Overview of Privacy-Leaking HTTP Headers

3.3.1 Referrer policy and the Referrer header

The **Referer** header is a request header that is intended to announce the context in which an HTTP request is initiated. It has been a part of the HTTP specification since the protocol's inception. The header is intended to aid in analytics, enabling origin hosts to know how end users access website resources. For example, it is commonly used for search engine optimization (SEO) purposes to enable origin hosts to know which search terms people are using to discover their pages—this can help with marketing efforts. Figure 3.1 illustrates how the **Referer** header is propagated to third-parties when an HTTP request is initiated. The header is also occasionally used to protect against cross-site request forgery (CSRF) by checking whether the **Referer** header matches an expected origin webpage.

In the current (December, 2017) W3 standard, there are four ways for an origin site to specify when and in what context a browser should set the **Referer** header:

1. An HTML meta header that takes the form: `<meta name="referrer">`. This applies a default policy for all content loaded on that document and also determines whether the **Referer** header is included with outgoing links
2. A **Referrer-Policy** response header for default policy
3. A referrer policy attribute to declare a policy for an individual content reference e.g. ``
4. Another (older) policy attribute on an individual content reference to explicitly declare that a **Referer** header should be omitted e.g. ``

Collectively, the settings defined by these methods define the *Referrer Policy* on a webpage. There are a variety of possible policies declared in the W3 specification:

1. `"no-referrer"`: No `Referer` header is ever sent.
2. `"no-referrer-when-downgrade"`: A `Referer` header containing the full URL is only sent to links also using HTTPS. *This is the default behavior, if no policy is otherwise specified.*
3. `"same-origin"`: A `Referer` header containing the full current URL is sent only when making same-origin requests, otherwise the header is omitted entirely.
4. `"origin"`: A `Referer` header is sent that only contains the root domain.
5. `"strict-origin"`: Same as `"origin"`, but the header is only sent if HTTPS is used.
6. `"origin-when-cross-origin"`: Sends the full URL for same-origin requests, otherwise only sends the root domain.
7. `"strict-origin-when-cross-origin"`: Same as `"origin-when-cross-origin"` but the header is only sent if HTTPS is used.
8. `"unsafe-url"`: `Referer` header is always sent with the full URL.
9. `""`: An empty string will result in the default policy being used i.e. `"no-referrer-when-downgrade"`

`Referer` headers and referrer policy are quite complicated for website administrators to manage. Further subtleties exist on the way these features are implemented—these are described in detail in the W3 specification. The default policy recommended by W3 is `"no-referrer-when-downgrade"`, which can put end users' privacy at risk by exposing browsing history and personally identifiable information (PII) contained in referral URLs.

3.3.2 Cross-Origin Resource Sharing (CORS) and the Origin header

The **Origin** header is a request header similar to the **Referer** header that is intended to announce the domain from which an HTTP request originated. It is often sent *in addition to* the **Referer** header and is used to authenticate cross-origin requests based on access control policies that either enforce or circumvent the same-origin rule enabled by default in all modern browsers. CORS was introduced to HTTP after the same-origin policy was implemented in browsers to enable safe sharing across domains.

The **Origin** header is generated by **XmlHttpRequests** (XHRs) in dynamic resource loads. This header can leak information about an end user's browsing habits, though to a lesser degree than the **Referer** header. However, in situations where a certain resource is deployed on a single page on a website, a request for that resource can uniquely identify a user as having visited that page—in the case where there are a few pages that embed a resource, this header can narrow down the possible origin pages to a very small subset.

There are many HTTP headers associated with CORS. Below are the request headers and their usage/interpretation:

1. **Origin**: sent by browser to indicate domain from which request originated
2. **Access-Control-Request-Method**: used when issuing a preflight request to let the server know which HTTP method will be used when the actual request is made
3. **Access-Control-Request-Headers**: used when issuing a preflight request to let the server know which HTTP headers will be used when the actual request is made

Below are the response headers and their usage/interpretation:

1. **Access-Control-Allow-Origin:** indicates which domains can make cross-origin requests
2. **Access-Control-Allow-Credentials:** indicates whether or not the response to the request can be exposed to the page
3. **Access-Control-Expose-Headers:** indicates which headers can be exposed as part of the response by listing their names
4. **Access-Control-Max-Age:** indicates how long the results of a preflight request can be cached
5. **Access-Control-Allow-Methods:** indicates which HTTP methods are allowed i.e. GET, POST, PUT
6. **Access-Control-Allow-Headers:** indicates which HTTP headers can be used when making the actual request (sent in preflight request)

3.4 Measuring Referrer Policy

Out of 89,795,315 requests made in the 2016 OpenWPM crawl of the top million websites, 85,242,413 (94.5%) contained a **Referer** header. This indicates that almost all requests leak all or part of their referring URLs to most third-parties in use on their pages.

We then examine the use of meta headers/response headers containing referrer policy. The results are given in Table 3.1. Most websites lack such a header (the feature was only proposed in 2015 and implemented in modern browsers shortly afterwards). We observed only 1,732 sites out of 899,009 (0.19%) utilizing this feature—of these sites 628 were using policies that leak URLs to third-party URLs

Table 3.1: Referrer policy usage in the top 1M Alexa-ranked websites

Referrer Policy	Count
<i>(No meta header present)</i>	897277
no-referrer-when-downgrade	467
origin-when-cross-origin	303
no-referrer	268
strict-origin-when-cross-origin	183
unsafe-url	156
same-origin	131
origin	118
strict-origin	47
no-referrer, strict-origin-when-cross-origin	19
origin-when-cross-origin, strict-origin-when-cross-origin	11
no-referrer, same-origin	9
<i>(Meta header present but empty)</i>	5
origin, unsafe-url	2
no-referrer-when-downgrade, strict-origin-when-cross-origin	2
default,origin-when-cross-origin,same-origin	1
no-referrer,origin-when-cross-origin	1
no-referrer,origin-when-cross-origin, strict-origin-when-cross-origin	1
no-referrer,same-origin	1
no-referrer,strict-origin-when-cross-origin	1
no-referrer-when-downgrade,strict-origin-when-cross-origin	1
same-origin, strict-origin-when-cross-origin	1
same-origin,strict-origin	1
<i>(Erroneous meta header present)</i>	3
TOTAL	899009

(no-referrer-when-downgrade, unsafe-url, and empty policy). Additionally, three sites contained erroneous meta headers that would most likely result in the default policy (no-referrer-when-downgrade) being assumed on most modern browsers. This means that only 1,101 sites out of 899,009 (0.12%) are by default protecting user privacy in requests being made.

Furthermore, we examine the deployment of referrer policy by website rank in Figure 3.2. The distribution is well-approximated by a power-law curve yielding a

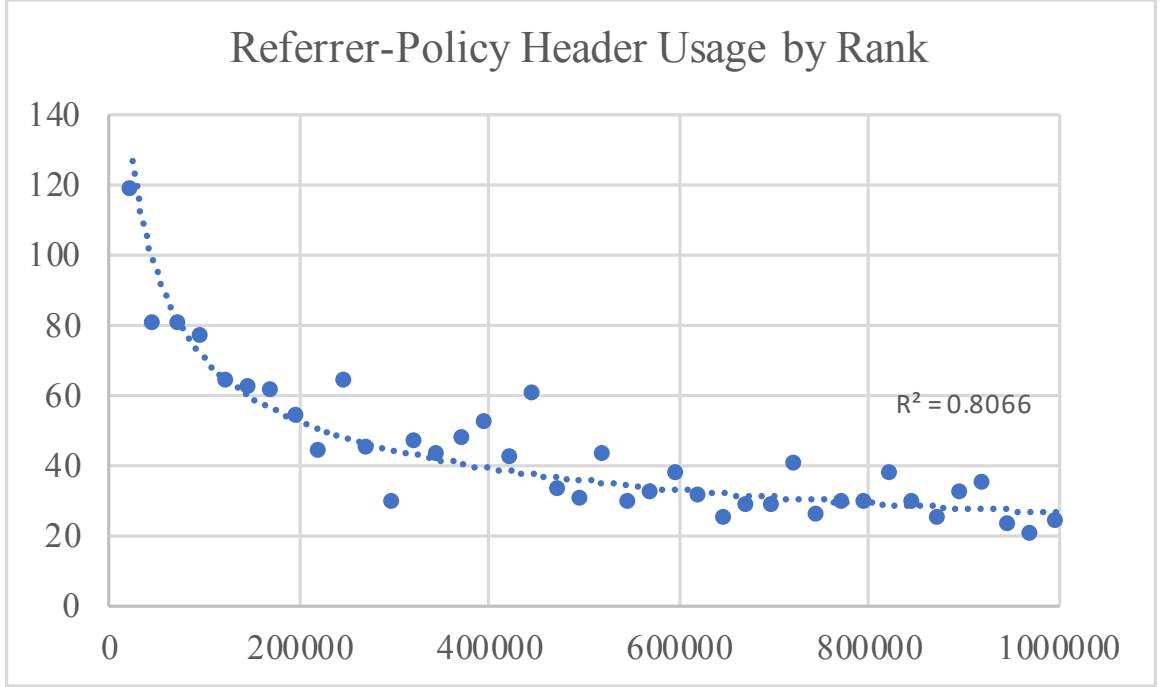


Figure 3.2: A plot of `Referer` header usage on root pages of the top million websites (binned by 25,000). The plot roughly follows power-law distribution, yielding a probability function: $\text{Prob}(\text{Referer Header Usage}) = 0.357(\text{Rank})^{-0.42}$ with $R^2 = 0.80659$

probability function:

$$\text{Prob}(\text{Meta Header Usage}) = 0.357(\text{Rank})^{-0.42}$$

$$(R^2 = 0.80659)$$

3.5 Measuring CORS Policy

Out of 89,795,315 requests made in the 2016 OpenWPM crawl of the top million websites, 3,020,427 contained an `Origin` header (3.36%). OpenWPM keeps track of the XHR requests made through dynamic loads.

We then examine the use of `Access-Control-Allow-Origin` headers containing referrer policy. The results are given in Table 3.2. We observed only 5,798 sites out

Table 3.2: CORS policy usage in the top 1M Alexa-ranked websites

CORS Policy	Count
<i>(No CORS header present)</i>	868803
* <i>(All origins allowed)</i>	24408
<i>(One or more origins specified)</i>	5798
TOTAL	899009

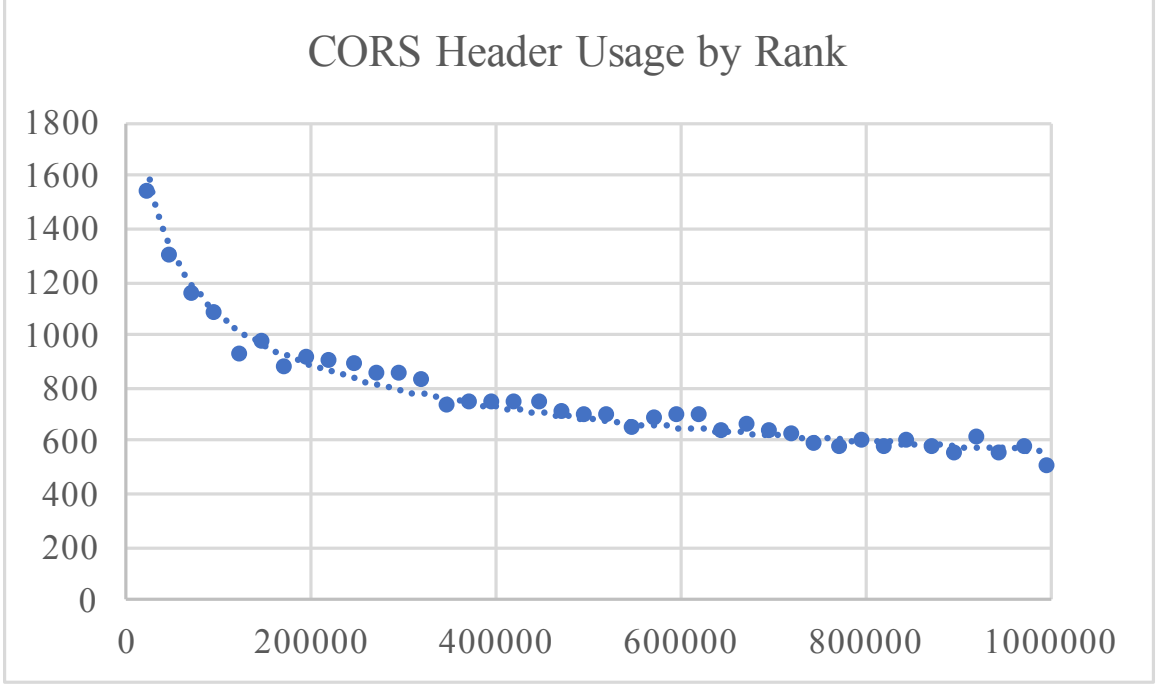


Figure 3.3: A plot of CORS header usage on root pages of the top million websites (binned by 25,000). The plot roughly follows power-law distribution, yielding a probability function: $\text{Prob}(\text{CORS Header Usage}) = 1.086(\text{Rank})^{-0.281}$ with $R^2 = 0.96861$

of 899,009 (0.645%) utilizing explicit origin-setting CORS policy. The rest implicitly leak origin information to the third-parties involved.

Furthermore, we examine the deployment of CORS policy by website rank in Figure 3.3. The distribution is well-approximated by a power-law curve yielding a probability function:

$$\text{Prob}(\text{CORS Header Usage}) = 1.086(\text{Rank})^{-0.281}$$

$$(R^2 = 0.96861)$$

3.6 Reforming Referrer Policy

We recommend that the default Referrer Policy in browsers be set to `no-referrer` or at least an `origin`-based setting as opposed to the current default (which is `no-referrer-with-downgrade`) that leaks the full URL to the receiving page. This protects user privacy in the most common scenarios and would prevent the attacks described earlier by Krishnamurthy et al. [33].

This solution is implemented in some browser extensions already. Keepa.com provides a browser extension called *Referer Control* that is available for both Mozilla Firefox¹ and Google Chrome² that allows a user to set the default referrer policy, and also enables control over referrer policy on domain-level, page-level, or resource-level basis. We also suggest more advanced solutions with greater privacy properties in our future work chapter.

3.7 Conclusion

In this section, we summarize and examine the privacy threat of cross-site metadata aggregation from third-party offload. We find through measurement that many third-party organizations have content deployed across a large fraction of the top-million websites, which in conjunction with weaknesses in default referrer policy/CORS policy, results in end-user privacy leakage. Specifically, our preliminary findings show that these third-parties have a global picture of Internet browsing down to the level of an individual. Notably, these passive privacy attacks do not make use of cookies for tracking.

¹<https://addons.mozilla.org/en-US/firefox/addon/referercontrol/>

²<https://chrome.google.com/webstore/detail/referer-control/hnkcfpcejkafcihlgbjoidoihckciin>

Chapter 4

Resource Heterogeneity and Traffic Fingerprinting Threats

4.1 Introduction

In the previous chapter, we showed that referrer policy on the Internet tends to leak the user’s browsing habits to third parties. The fix we proposed for referrer policy was very simple—change the default policy to one that enables greater privacy for end users, such as `origin` or `no-referrer`. This change would only require a simple browser modification and could be easily deployed on the top browsers.

However, the `Referer` header is not always necessary to determine which page a request originated from. Different webpages make unique resource requests to third parties and these resource request patterns can be leveraged by the third parties to discover from which page the request originated. We coin this phenomenon *resource heterogeneity*—this chapter is focused on its measurement and characterization. In the previous chapter, we discussed related work on traffic fingerprinting based on interception of (encrypted) network traffic between a client and a server. In this case, however, the CDNs behave as men-in-the-middle and have unencrypted access to the

requests being made to their servers. We demonstrate that it is likely that a CDN could use massive web crawls to fingerprint websites and determine where requests for a certain set of resources originated.

4.2 The CommonCrawl Dataset

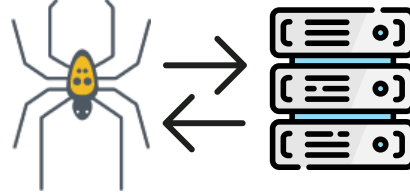
We utilize data from the CommonCrawl, a massive web archive of pages across millions of domains based on large-scale web crawling. The CommonCrawl has some disadvantages when compared to crawls like OpenWPM—it involves purely static web crawling (hence missing the many dynamic third-party requests that are generated) and has relatively low coverage of the top million Alexa domains (about 35%), partially due to its adherence to the `robots.txt` protocol. The lack of dynamic third-party requests is also significant in that the CommonCrawl gives us much lower resource heterogeneity numbers than we would expect to see with a dynamic crawl.

However, the CommonCrawl does have some key advantages over dynamic crawls which we believe make it viable for the fingerprinting attacks we envision. It contains a much larger and deeper set of pages—for example, the March 2018 crawl employed in this study contained data from 3.2 billion web page crawls with over 250+ TiB of uncompressed content, crawled between March 17th and 25th. Additionally, this single crawl contains 800 million new URLs, not contained in any crawl archive before. Given the fact that the infrastructure to manipulate this data is quite significant (Amazon Web Services hosts the data and provides mechanisms for straightforward analysis of it), we believe that the CommonCrawl is an excellent resource for resource request fingerprinting.

1. Retrieve Alexa Top Sites



2. Use CommonCrawl to retrieve
to root domain



3. Extract links

``

A black curly brace underlines the URL "http://www.example.com" in the HTML tag.

4. Compare resource sets

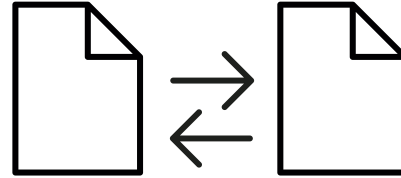


Figure 4.1: The steps involved in cross-comparison of CDN resource requests.

4.3 Methodology

We set up a CommonCrawl index on an AWS EC2 instance for March 2018 and ran our analysis on the top million Alexa domains. We crawled the root pages of each domain and extracted the lists of external (i.e. third-party) static resources that would be requested as a result. We then cross-compared these lists to determine how often the static resource set differed and would be susceptible to fingerprinting. A representation of this methodology is given in Figure 4.1.

In our analysis, we consider the number of websites that would have the same resource request profile based on the static resources parsed from HTML files on their root pages. We extracted the `src` attributes from every tag and collected them into a database. This enabled us to construct k -anonymous sets of websites, where each resource set could be associated with any one of k possible origin domains in that

set. **Resource heterogeneity** is the idea that the average k for these k -anonymous sets is small—in other words, most websites have unique resource request patterns

4.4 Results and Analysis

Out of the 359,097 top-million Alexa websites that we were able to obtain crawl data for from the CommonCrawl, 167,690 (46.7%) had no explicit static requests to external resources, 172,884 (48.1%) had completely unique external resource sets, and the remaining 18,523 (5.2%) had overlapping external resource sets with one or more other websites. The most common resource sets are given in Table 4.1 and are mostly API/advertising script requests.

We also present the distribution of k -anonymous set quantity by set size in Figure 4.2. Another (possibly more useful) plot is given in Figure 4.3, which indicates the number of sites in a k -anonymous set of given set size. The plot takes an interesting shape—we notice a somewhat parabolic distribution here which indicates that there are a large number of sites that have a small anonymity set and also a significant number of sites that have a large anonymity set. A majority of the websites have k -anonymous sets of size 1, meaning they generate completely distinct resource requests from the other top-million domains—also many of the remaining sets have size less than 10. This means that in most cases, the external resource requests generated by many top-million pages are distinct and can easily be profiled by the third parties to which the requests are made. This is problematic, in that it enables easy characterization of user browsing habits.

4.5 Conclusion

We make concrete measurements of the observation that different webpages have different external resource request patterns. We find that a majority of webpages

Table 4.1: Most popular external resource sets (static) found with CommonCrawl on top million websites

External Resource Set	Number of Sites
(pagead2.googlesyndication.com/pagead/js/adsbygoogle.js)	2554
(pagead2.googlesyndication.com/pagead/show_ads.js)	452
(assets.tumblr.com/client/prod/standalone/error-pages/index.build.js?_v=4caca..., assets.tumblr.com/languages/errors.js?_v=2d965...)	327
(www.google-analytics.com/urchin.js)	267
(www.google.com/recaptcha/api.js)	223
(ajax.googleapis.com/ajax/libs/jquery/1.11.0/jquery.min.js)	194
(www.google-analytics.com/analytics.js)	190
(ajax.googleapis.com/ajax/libs/jquery/1.11.1/jquery.min.js)	186
(www.google.com/recaptcha/api/fallback?k=6LfBi...)	185
(translate.google.com/translate_a/element.js?cb=googleTranslateElementInit)	184
(platform.twitter.com/widgets.js)	160
(ajax.googleapis.com/ajax/libs/jquery/1.10.2/jquery.min.js)	152
(ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js)	127
(pagead2.googlesyndication.com/pagead/js/adsbygoogle.js, pagead2.googlesyndication.com/pagead/show_ads.js)	126
(www.google.com/jsapi)	122
(apis.google.com/js/plusone.js)	120
(ajax.googleapis.com/ajax/libs/jquery/1.9.1/jquery.min.js)	118
(ajax.googleapis.com/ajax/libs/jquery/1.7.1/jquery.min.js)	112
(apis.google.com/js/platform.js)	111
(www.googletagservices.com/tag/js/gpt.js)	109
(ajax.googleapis.com/ajax/libs/jquery/1.8.3/jquery.min.js)	103
(ajax.googleapis.com/ajax/libs/jquery/1.7.2/jquery.min.js)	91
(ajax.googleapis.com/ajax/libs/jquery/1.11.2/jquery.min.js)	88
(cdn.onesignal.com/sdks/OneSignalSDK.js, pagead2.googlesyndication.com/pagead/js/adsbygoogle.js)	86
(www.adobe.com/images/shared/download_buttons/get_flash_player.gif)	86
(ajax.googleapis.com/ajax/libs/jquery/1.8.2/jquery.min.js)	86
(s0.wp.com/wp-content/js/devicepx-jetpack.js?ver=201812, stats.wp.com/e-201812.js)	85
(apis.google.com/js/platform.js, pagead2.googlesyndication.com/pagead/js/adsbygoogle.js)	84
(cdn.onesignal.com/sdks/OneSignalSDK.js)	82
(w.sharethis.com/button/buttons.js)	75

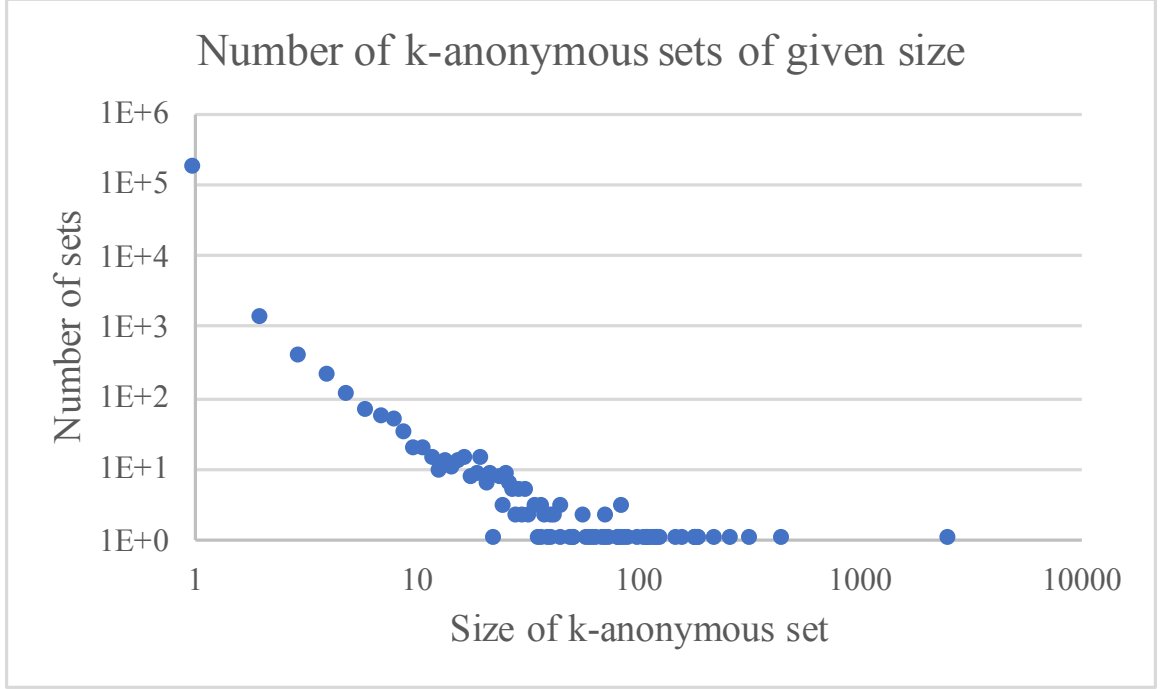


Figure 4.2: The distribution of k -anonymous set quantity by set size. We see the existence of a Zipf-like pattern in this log-log plot.

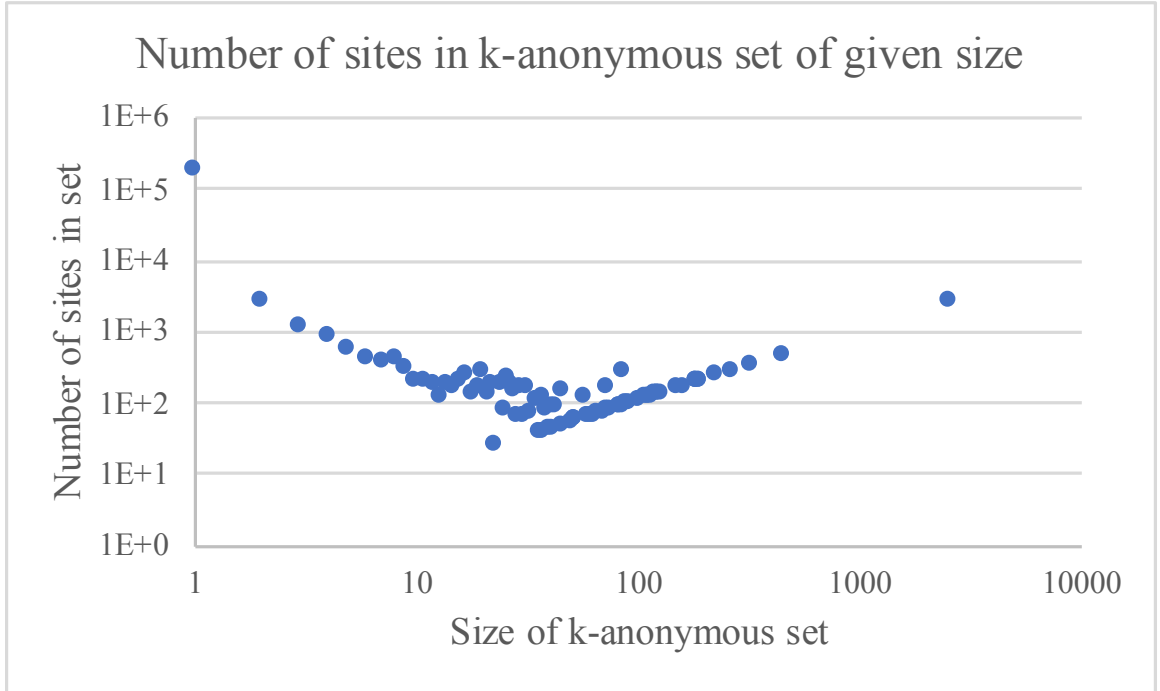


Figure 4.3: The distribution of number of sites in a given k -anonymous set by set size. This figure is the same as Figure 4.2 except each data point is multiplied by the size at that point.

have unique patterns that make for easy fingerprinting by the third parties involved. This phenomenon of *resource heterogeneity* makes the anonymous use of third parties for content delivery very difficult, even in the absence of giveaway headers such as `Referer` and `Origin`. Our work in this chapter greatly motivates the need for robust privacy-preserving content delivery systems.

Chapter 5

Future Work and Conclusion

5.1 Future Projects

5.1.1 Deeper crawls and more intricate measurements

In this study, we only employ crawls on root webpages of the top million sites, and we do not crawl sites that require federated login to enable their full functionality. Studying how deeper crawls affect our measurements will be necessary for demonstrating practical privacy-preserving tools in the future. Additionally, it remains to be seen to what extent modifying the way referrer policies and CORS policies are handled by browsers impacts the usability of the modern Internet. It would also be interesting to investigate how caching in modern browsers/ISPs would interact with the sorts of measurements we perform.

5.1.2 Monitoring third-party information leakage via real-time measurement framework

By understanding how users' browsing behavior results in diffusion of privacy across the biggest third-parties, they can adapt their Internet habits as they see fit, to

result in reduced loss of privacy. We hope to build and release a tool of this nature to monitor and report third-party information leakage with/without cookies while browsing the Internet.

5.1.3 Investigating offload to multiple third-parties by origin hosts, splitting user traffic amongst them

While this will not eliminate privacy threats from CDNs and other third-parties entirely, it can prevent aggregation of complete browsing information in a single third-party. Origin hosts can also strategically split traffic such that end users' privacy is maximized.

5.1.4 Splitting traffic between multiple third-party options provided to end users by origin hosts

This proposal for mitigating third-party browser history aggregation is similar to the one described above, except the origin hosts provide multiple options to the end user, who can then decide how to split their traffic such that their personal privacy needs are met. Similar to the preceding strategy, this technique does not eliminate the privacy threats—it only reduces their severity.

5.1.5 Investigating the feasibility of proxies/Onion routing to access CDN content

Using a proxy/Tor to access CDN content would protect privacy by leveraging an anonymous identity (i.e. IP address) shared by multiple users to access CDN content. However, this approach may introduce performance degradation due to additional latency and limited throughput offered by anonymity overlays, negating the purpose

of using CDNs in the first place. It remains to be seen if privacy overlay networks can provide a practical solution for content delivery with low latency.

5.1.6 P2P content delivery

There has been much interest in establishing peer-to-peer content delivery for high availability, robustness, and low latency benefits that may be reaped. P2P content delivery could enable users to trade off privacy with performance by routing requests through peers for anonymity. This approach has been studied by Edmundson et al. in *OCDN: Oblivious Content Distribution Networks*, which describes an approach to content delivery where the CDN is oblivious to the content it hosts [42]. This approach is inspired by onion routing, variable-path P2P routing, and cryptographic techniques for secure multi-party computation and storage.

5.1.7 End users can access CDN content using multiple IP addresses

For scenarios where CDNs are uniquely identifying clients via source IP addresses, using multiple IP addresses provides an end user with multiple identities. By splitting traffic across these identities while browsing, an end user can reduce privacy leakage from using a single identity—given enough possible identities (i.e., IP addresses), the privacy threat posed by CDNs could be eliminated altogether. Given the large availability of IPv6 addresses, this solution may be practical for CDNs with IPv6 support—the proposal by Han et al. for *Expressive Privacy Control with Pseudonyms* is an example of prior work of this nature [43].

5.1.8 Investigating CDNi and its user privacy implications

Content Delivery Network interconnection (CDNi) is an emerging trend in modern content delivery, where CDNs will deliver content on behalf of one another. Interconnected CDNs offer many benefits, such as footprint extension, reduced infrastructure costs, higher availability, etc., for content service providers (CSPs), CDNs, and end users. However, one might be concerned that this could lead to collusion amongst CDNs for end users' browsing metadata, including PII associated with browsing history. Measuring CDNi on the top million websites through DNS scans could provide insight into whether privacy threats exist in this model. The techniques would involve investigating the many-to-many graph of forward DNS and reverse DNS relationships from different vantage points in the Internet (perhaps using PlanetLab infrastructure) and seeing if CDNs are observed to be negotiating responsibility for content delivery in different contexts.

5.1.9 Automated CDN whitelist generation

In using the manual whitelist from `webpagetest.org`, we noticed that manual whitelisting produces low coverage for smaller/less well-known CDNs. We believe that large-scale automated web crawls can assist in the mapping of hostnames to the CDNs responsible for them. A future direction would involve the use of heuristics for determining CDNs from fully-instrumented web crawls.

5.2 Conclusion

In conclusion, this thesis has thoroughly examined passive privacy threats due to the widespread use of Content Delivery Networks on the Internet. We demonstrate mechanisms for determining CDN usage, and show that there are many privacy threats that have been largely overlooked—namely, the prevalence of requests to many top

CDNs, the lack of privacy-protecting referrer policy, and resource heterogeneity, which enables greater ease in third-party fingerprinting. Ultimately, we show that CDN privacy threats are pervasive and largely overlooked in the modern day Internet infrastructure and we hope that this thesis enables and encourages more research on passive privacy attacks from third-parties.

Appendix A

CNAME Whitelist for Small-Scale Measurement

The whitelist used for CNAME matching is provided below. The left entry is the network location to be matched (via string search) and the right entry is the corresponding CDN:

```
.akamai.net: Akamai
.akamaized.net: Akamai
.akamaiedge.net: Akamai
.akamaihd.net: Akamai
.edgesuite.net: Akamai
.edgekey.net: Akamai
.srip.net: Akamai
.akamaitechnologies.com: Akamai
.akamaitechnologies.fr: Akamai
.tl88.net: Akamai
.llnwd.net: Limelight
```

edgecastcdn.net: Edgecast
.systemcdn.net: Edgecast
.transactcdn.net: Edgecast
.v1cdn.net: Edgecast
.v2cdn.net: Edgecast
.v3cdn.net: Edgecast
.v4cdn.net: Edgecast
.v5cdn.net: Edgecast
hwcdn.net: Highwinds
.simplecdn.net: Simple CDN
.instacontent.net: Mirror Image
.footprint.net: Level 3
.fpbns.net: Level 3
.insnw.net: Instart Logic
.inscname.net: Instart Logic
.internapcdn.net: Internap
.cloudfront.net: Amazon CloudFront
.netdna-cdn.com: NetDNA
.netdna-ssl.com: NetDNA
.netdna.com: NetDNA
.kxcdn.com: KeyCDN
.cotcdn.net: Cotendo CDN
.cachefly.net: Cachefly
bo.lt: B0.LT
.cloudflare.com: Cloudflare
.afxcdn.net: afxcdn.net
.lxdns.com: ChinaNetCenter

.wscdns.com: ChinaNetCenter
.wscloudcdn.com: ChinaNetCenter
.ourwebpic.com: ChinaNetCenter
.att-dsa.net: AT&T
.vo.msecnd.net: Microsoft Azure
.azureedge.net: Microsoft Azure
.voxcdn.net: VoxCDN
.bluehatnetwork.com: Blue Hat Network
.swiftcdn1.com: SwiftCDN
.cdngc.net: CDNNetworks
.gccdn.net: CDNNetworks
.panthercdn.com: CDNNetworks
.fastly.net: Fastly
.fastlylb.net: Fastly
.nocookie.net: Fastly
.mirror-image.net: Mirror Image
.yottaa.net: Yottaa
.cubecdn.net: cubeCDN
.cdn77.net: CDN77
.cdn77.org: CDN77
.incapdns.net: Incapsula
.bitgravity.com: BitGravity
.r.worldcdn.net: OnApp
.r.worldssl.net: OnApp
.ngenix.net: NGENIX
.pagerain.net: PageRain
.ccgslb.com: ChinaCache

cdn.sfr.net: SFR
.azioncdn.net: Azion
.azioncdn.com: Azion
.azion.net: Azion
.cdncloud.net.au: MediaCloud
.rncdn: Reflected Networks
.cdnsun.net: CDNsun
.mncdn.com: Medianova
.mncdn.net: Medianova
.mncdn.org: Medianova
cdn.jsdelivr.net: jsDelivr
.nyiftw.net: NYI FTW
.nyiftw.com: NYI FTW
.resrc.it: ReSRC.it
.zenedge.net: Zenedge
.lswcdn.net: LeaseWeb CDN
.lswcdn.eu: LeaseWeb CDN
.revcn.net: Rev Software
.revdn.net: Rev Software
.caspowa.com: Caspowa
.rlcdn.com: Reapleaf
.wp.com: WordPress
.aads1.net: Aryaka
.aads-cn.net: Aryaka
.aads-cng.net: Aryaka
.squixa.net: section.io
.bisongrid.net: Bison Grid

.cdn.gocache.net: GoCache
.hiberniacdn.com: HiberniaCDN
.cdntel.net: Telenor
.raxcdn.com: Rackspace
.unicorncdn.net: UnicornCDN
.optimalcdn.com: Optimal CDN
.hosting4cdn.com: Hosting4CDN
.netlify.com: Netlify

.taobao.com: Alibaba
.tbcache.com: Alibaba
.tbcdn.cn: Alibaba
.taobaocdn.com: Alibaba
.alicdn.: Alibaba
.cdnetworks.net: CDNNetworks
.chinacache.net: ChinaCache
.akadns.: Akamai
.yunjiasu-cdn.net: Cloudflare
.cloudflare.net: Cloudflare

Appendix B

Examining CDN Ownership of Tor Relays

CDN ownership of Tor relays could compromise privacy of users in the Tor network. If CDNs can access traffic at both ends of the Tor network (between client and Tor network, and between Tor network and destination), then they can perform end-to-end timing correlation attacks to deanonymize users of the Tor network [44]. CDNs could also serve as AS-level adversaries and perform RAPTOR attacks on traffic in the Tor network [45]—lack of diversity in Tor relays has been already been cited as a possible weakness of the Tor network by Feamster and Dingleline [46]. Here, we use measurement techniques described earlier to determine whether such attacks might be viable, namely (1) ASN to organization mapping and (2) RDAP lookups.

In Table B.1, we describe AS ownership of Tor relays as a whole. Table B.2 describes ownership of Tor relays at the granularity of RDAP records based on the contact name, filtered to only contain organization names. Reverse DNS lookups showed only 10 relays associated with a CDN in our whitelist, so we did not include this method in our analysis.

Table B.1: Tor relay ASN Lookups: Organization Counts

AS Organization	Count
OVH SAS	490
ONLINE S.A.S.	359
Digital Ocean, Inc.	302
Hetzner Online GmbH	299
Linode, LLC	195
Deutsche Telekom AG	168
Liberty Global Operations B.V.	105
Comcast Cable Communications, LLC	104
Choopa, LLC	99
Time Warner Cable Internet LLC	53
Host Europe GmbH	50
Contabo GmbH	48
Free SAS	45
Amazon.com, Inc.	43
Vodafone Kabel Deutschland GmbH	43
MCI Communications Services, Inc.	42
myLoc managed IT AG	42
LeaseWeb Netherlands B.V.	40
PJSC Rostelecom	38
M247 Ltd	38
ITL Company	37
Vodafone GmbH	32
British Telecommunications PLC	31
1&1 Versatel Deutschland GmbH	30
Quintex Alliance Consulting	30
QuadraNet, Inc	30
Telenor Norge AS	29
KW Datacenter	29
Xs4all Internet BV	27
Orange S.A.	27
Bahnhof Internet AB	27
netcup GmbH	27
Microsoft Corporation	26
Keyweb AG	26
Makonix SIA	26

Table B.2: Tor relay RDAP Lookups: Organization Counts

RDAP Organization	Count
Hetzner Online GmbH	60
Deutsche Telekom	51
DigitalOcean, LLC	49
Linode Network Operations	36
Comcast Cable Communications, LLC	36
Vultr Holdings, LLC	27
OVH Hosting, Inc.	26
Unitymedia Administration	18
Japan Network Information Center	18
RIP Mean	17
Proxad	16
Liberty Global RIPE DBM	15
IRT-JPNIC-JP	15
Contabo	14
Kabel Deutschland RIPE	13
Comcast Cable Communications, Inc.	12
KW Datacenter	12
HEG	12
Time Warner Cable Internet LLC	12
myLoc	12
MCI Communications Services, Inc. d/b/a Verizon Business	11
Verizon Internet Services	11
KDCTRL	11
Amazon EC2	11
JSC Rostelecom	11
Technologies UUnet Technologies, Inc.	11
mediaWays	10
Charter Communications	10
Datasource AG	10
THCServers	10
LeaseWeb Netherlands	10
Google LLC	10
Versatel	10
Digital Ocean Inc	9
Vodafone Germany	9
Comcast IP Services, L.L.C.	8
XS4ALL Internet	8
GRnet	8
Netcup	8
France Telecom	8

Bibliography

- [1] Balachander Krishnamurthy, Craig Wills, and Yin Zhang. On the use and performance of content distribution networks. In *Proceedings of the 1st ACM SIGCOMM Workshop on Internet Measurement*, pages 169–182. ACM, 2001.
- [2] Limin Wang, KyoungSoo Park, Ruoming Pang, Vivek S Pai, and Larry L Peterson. Reliability and security in the CoDeeN content distribution network. In *USENIX Annual Technical Conference, General Track*, pages 171–184, 2004.
- [3] Jinjin Liang, Jian Jiang, Haixin Duan, Kang Li, Tao Wan, and Jianping Wu. When HTTPS meets CDN: A case of authentication in delegated service. In *Security and privacy (sp), 2014 IEEE symposium on*, pages 67–82. IEEE, 2014.
- [4] Harrison Weber. How the NSA & FBI made Facebook the perfect mass surveillance tool. VentureBeat Online Article, May 2014.
- [5] Roger Dingledine, Nick Mathewson, and Paul Syverson. Tor: The second-generation onion router. Technical report, Naval Research Lab Washington DC, 2004.
- [6] Steven Englehardt and Arvind Narayanan. Online tracking: A 1-million-site measurement and analysis. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pages 1388–1401. ACM, 2016.
- [7] Aniko Hannak, Gary Soeller, David Lazer, Alan Mislove, and Christo Wilson. Measuring price discrimination and steering on E-commerce web sites. In *Proceedings of the 2014 conference on internet measurement conference*, pages 305–318. ACM, 2014.
- [8] Kate Kaye. The \$24 Billion Data Business That Telcos Don’t Want to Talk About. Advertising Age Online Article, October 2015.
- [9] US Congress. Senate. committee on the judiciary, subcommittee on constitutional amendments. In *Hearings Before the Subcommittee on Constitutional Amendments of the Committee of the Judiciary United States Senate Ninety-First Congress Second Session on SJ Res. 7, SJ Res. 19, SJ Res. 32, SJ Res. 34, SJ Res. 38, SJ Res. 73, SJ Res. 73, SJ Res. 87, SJ Res. 102, SJ Res. 105, SJ Res. 141, SJ Res.*, volume 147, pages 16–17.

- [10] Cheng Huang, Angela Wang, Jin Li, and Keith W Ross. Measuring and evaluating large-scale CDNs. In *ACM Internet Measurement Conference*, 2008.
- [11] Balachander Krishnamurthy and Craig E Wills. Generating a privacy footprint on the Internet. In *Proceedings of the 6th ACM SIGCOMM conference on Internet measurement*, pages 65–70. ACM, 2006.
- [12] Balachander Krishnamurthy, Delfina Malandrino, and Craig E Wills. Measuring privacy loss and the impact of privacy protection in web browsing. In *Proceedings of the 3rd symposium on Usable privacy and security*, pages 52–63. ACM, 2007.
- [13] Balachander Krishnamurthy and Craig Wills. Privacy diffusion on the web: a longitudinal perspective. In *Proceedings of the 18th international conference on World wide web*, pages 541–550. ACM, 2009.
- [14] Jessica Su, Ansh Shukla, Sharad Goel, and Arvind Narayanan. De-anonymizing web browsing data with social networks. In *Proceedings of the 26th International Conference on World Wide Web*, pages 1261–1269. International World Wide Web Conferences Steering Committee, 2017.
- [15] Richard Chow and Philippe Golle. Faking contextual data for fun, profit, and privacy. In *Proceedings of the 8th ACM workshop on Privacy in the electronic society*, pages 105–108. ACM, 2009.
- [16] Vincent Toubiana, Lakshminarayanan Subramanian, and Helen Nissenbaum. TrackMeNot: Enhancing the privacy of web search. *arXiv preprint arXiv:1109.4677*, 2011.
- [17] Antawan Holmes and Marc Kellogg. Automating functional tests using selenium. In *Agile Conference, 2006*, pages 6–pp. IEEE, 2006.
- [18] Rick Viscomi, Andy Davies, and Marcel Duran. *Using WebPageTest: Web Performance Testing for Novices and Power Users.* ” O’Reilly Media, Inc.”, 2015.
- [19] Deepak Kumar, Zane Ma, Zakir Durumeric, Ariana Mirian, Joshua Mason, J. Alex Halderman, and Michael Bailey. Security challenges in an increasingly tangled web. In *Proceedings of the 26th International Conference on World Wide Web*, 2017.
- [20] Devdatta Akhawe, Francois Marier, Frederik Braun, and Joel Weinberger. Sub-resource integrity. *W3C working draft, W3C, July*, 2015.
- [21] Gunes Acar, Christian Eubank, Steven Englehardt, Marc Juarez, Arvind Narayanan, and Claudia Diaz. The web never forgets: Persistent tracking mechanisms in the wild. In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*, pages 674–689. ACM, 2014.

- [22] Gunes Acar, Marc Juarez, Nick Nikiforakis, Claudia Diaz, Seda Gürses, Frank Piessens, and Bart Preneel. Fpdetective: dusting the web for fingerprinters. In *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security*, pages 1129–1140. ACM, 2013.
- [23] Alessandro Finamore, Vinicius Gehlen, Marco Mellia, Maurizio M Munafo, and Saverio Nicolini. The need for an intelligent measurement plane: The example of time-variant CDN policies. In *Telecommunications Network Strategy and Planning Symposium (NETWORKS), 2012 XVth International*, pages 1–6. IEEE, 2012.
- [24] Jonathan R Mayer and John C Mitchell. Third-party web tracking: Policy and technology. In *Security and Privacy (SP), 2012 IEEE Symposium on*, pages 413–427. IEEE, 2012.
- [25] Suqi Liu, Ian Foster, Stefan Savage, Geoffrey M Voelker, and Lawrence K Saul. Who is. com?: Learning to parse whois records. In *Proceedings of the 2015 ACM Conference on Internet Measurement Conference*, pages 369–380. ACM, 2015.
- [26] A Newton and S Hollenbeck. Json responses for the registration data access protocol (rdap). Technical report, 2015.
- [27] WebPageTest. WPO Foundation: cdn.h. GitHub.
- [28] Xue Cai, John Heidemann, Balachander Krishnamurthy, and Walter Willinger. Towards an as-to-organization map. In *Proceedings of the 10th ACM SIGCOMM conference on Internet measurement*, pages 199–205. ACM, 2010.
- [29] Frank Cangialosi, Taejoong Chung, David Choffnes, Dave Levin, Bruce M Maggs, Alan Mislove, and Christo Wilson. Measurement and analysis of private key sharing in the https ecosystem. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pages 628–640. ACM, 2016.
- [30] J Eisinger and E Stark. Referrer policy—editors draft, 28 march 2016. w3c. mar. 2016.
- [31] Dongseok Jang, Ranjit Jhala, Sorin Lerner, and Hovav Shacham. An empirical study of privacy-violating information flows in javascript web applications. In *Proceedings of the 17th ACM conference on Computer and communications security*, pages 270–283. ACM, 2010.
- [32] Balachander Krishnamurthy and Craig E Wills. On the leakage of personally identifiable information via online social networks. In *Proceedings of the 2nd ACM workshop on Online social networks*, pages 7–12. ACM, 2009.
- [33] Balachander Krishnamurthy, Konstantin Naryshkin, and Craig Wills. Privacy leakage vs. protection measures: the growing disconnect. In *Proceedings of the Web*, volume 2, pages 1–10, 2011.

- [34] Andriy Panchenko, Lukas Niessen, Andreas Zinnen, and Thomas Engel. Website fingerprinting in onion routing based anonymization networks. In *Proceedings of the 10th annual ACM workshop on Privacy in the electronic society*, pages 103–114. ACM, 2011.
- [35] Jamie Hayes and George Danezis. k-fingerprinting: A robust scalable website fingerprinting technique. In *USENIX Security Symposium*, pages 1187–1203, 2016.
- [36] Tao Wang and Ian Goldberg. Improved website fingerprinting on Tor. In *Proceedings of the 12th ACM workshop on Workshop on privacy in the electronic society*, pages 201–212. ACM, 2013.
- [37] Sai Teja Peddinti and Nitesh Saxena. On the privacy of web search based on query obfuscation: a case study of TrackMeNot. In *International Symposium on Privacy Enhancing Technologies Symposium*, pages 19–37. Springer, 2010.
- [38] Rami Al-Rfou, William Jannen, and Nikhil Patwardhan. TrackMeNot-so-good-after-all. *arXiv preprint arXiv:1211.0320*, 2012.
- [39] Vijay Kumar Adhikari, Yang Guo, Fang Hao, Matteo Varvello, Volker Hilt, Moritz Steiner, and Zhi-Li Zhang. Unreeling Netflix: Understanding and improving multi-CDN movie delivery. In *INFOCOM, 2012 Proceedings IEEE*, pages 1620–1628. IEEE, 2012.
- [40] Vijay Kumar Adhikari, Yang Guo, Fang Hao, Volker Hilt, and Zhi-Li Zhang. A tale of three CDNs: An active measurement study of Hulu and its CDNs. In *Computer Communications Workshops (INFOCOM WKSHPS), 2012 IEEE Conference on*, pages 7–12. IEEE, 2012.
- [41] Vijay K Adhikari, Yang Guo, Fang Hao, Volker Hilt, Zhi-Li Zhang, Matteo Varvello, and Moritz Steiner. Measurement study of Netflix, Hulu, and a tale of three CDNs. *IEEE/ACM Transactions on Networking (TON)*, 23(6):1984–1997, 2015.
- [42] Anne Edmundson, Paul Schmitt, Nick Feamster, and Jennifer Rexford. Ocdn: Oblivious content distribution networks. *arXiv preprint arXiv:1711.01478*, 2017.
- [43] Seungyeop Han, Vincent Liu, Qifan Pu, Simon Peter, Thomas Anderson, Arvind Krishnamurthy, and David Wetherall. Expressive privacy control with pseudonyms. *ACM SIGCOMM Computer Communication Review*, 43(4):291–302, 2013.
- [44] Aaron Johnson, Chris Wacek, Rob Jansen, Micah Sherr, and Paul Syverson. Users get routed: Traffic correlation on Tor by realistic adversaries. In *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security*, pages 337–348. ACM, 2013.

- [45] Yixin Sun, Anne Edmundson, Laurent Vanbever, Oscar Li, Jennifer Rexford, Mung Chiang, and Prateek Mittal. RAPTOR: Routing attacks on privacy in Tor. In *USENIX Security Symposium*, pages 271–286, 2015.
- [46] Nick Feamster and Roger Dingledine. Location diversity in anonymity networks. In *Proceedings of the 2004 ACM workshop on Privacy in the electronic society*, pages 66–76. ACM, 2004.